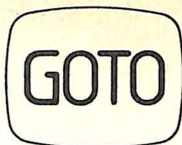
William Turner

APPLE II APPLE IIe

come si usa e cosa sa fare



CLUB DEL LIBRO
FRATELLI MELITA



Collana diretta da Giulio Palumbo

4

Prima edizione: dicembre 1984

Stampato presso le «A.C. Grafiche», Città di Castello
dalla Casa Editrice Anthropos s.r.l., Roma
per conto della Casa del Libro F.lli Melita s.n.c., La Spezia

William Turner

Scoprite il vostro Apple II ed Apple IIe

Traduzione di Ornella Imbastari

CLUB DEL LIBRO FRATELLI MELITA

Indice

Capitolo 1
Introduzione all'Apple II

Cos'è l'Apple II?

L'Apple II è un elaboratore (attualmente è un elaboratore della serie che comprende l'Apple II +, l'Apple IIe, l'Apple III ed il Lisa). È normalmente chiamato un microelaboratore (e a volte un Personal od un piccolo elaboratore gestionale), poiché è estremamente piccolo a confronto dei precedenti ed anche perché il suo «cuore» elettronico è un microprocessore. Come potete vedere dalla figura 1.1, l'Apple II sembra una cassetta con una tastiera, piuttosto che una normale macchina da scrivere. È normalmente usato con almeno uno, e spesso due, «drive» che si presentano come scatole separate. Per l'uso il sistema ha bisogno di uno schermo per la lettura. Questo può essere qualsiasi tipo di visore od uno proprio dell'Apple.



Fig. 1.1 L'Apple II

All'interno dell'Apple, come mostra la figura 1.2 vi sono dei circuiti integrati, detti *chips*; uno di questi contiene il microprocessore. Gli altri formano la memoria dell'elaboratore e possono memorizzare informazioni. All'inizio non serve occuparsi dell'interno dell'elaboratore. I circuiti elettronici e le apparecchiature che permettono il lavoro dell'Apple II sono affascinanti, ma non è necessaria una loro approfondita conoscenza per l'uso dell'elaboratore e questo libro tratta l'uso dell'Apple II.

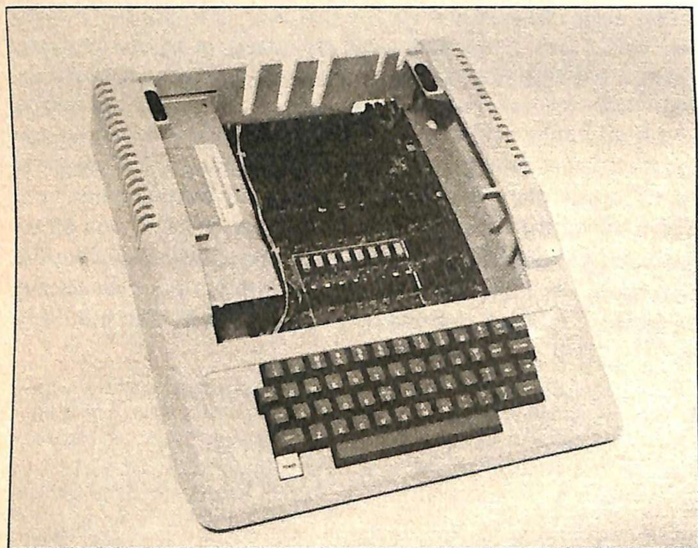


Fig. 1.2 L'interno dell'Apple II

Prima di andare avanti, tuttavia, vale la pena di sottolineare i seguenti punti: per molti versi l'Apple II è inteso non tanto come un elaboratore, quanto come un sistema di elaboratori. Negli anni passati è stato l'unico microelaboratore di enorme successo nel mondo. Centinaia di società devono

la loro esistenza a questo successo. Alcune esistono unicamente per scrivere pacchetti di software per l'Apple. Altre esistono per progettare, produrre e distribuire accessori che rendono l'Apple più veloce, più potente od addirittura in grado di svolgere altri compiti. Nella sola Italia, naturalmente, vi sono dozzine di altre società che esistono esclusivamente per vendere l'Apple.

Per tutti questi interessi commerciali non è sorprendente che vi siano dozzine di differenti versioni del sistema Apple II. In effetti si potrebbe dire anche che il tipo di Apple che voi comprate dipende dal motivo dell'acquisto e dal commerciante che ve lo fornisce. Per esempio se comprate l'Apple come «*word processor*» (trattamento dei testi), lo acquisterete probabilmente con una stampante chiamata «*stampante a margherita*». Se lo comprate per visualizzare grafici probabilmente comprerete una scheda aggiuntiva. Questo è il motivo per cui diciamo che l'Apple sia un sistema e non solo un elaboratore.

La tastiera è il principale componente dell'Apple II e serve per dialogare con l'elaboratore. Vi si possono rappresentare con semplicità i comandi e l'informazione da immagazzinare. Poiché la tastiera è posta come in una macchina da scrivere, un buon dattilografo può scrivere quasi con la stessa velocità che su una ordinaria macchina da scrivere. Per questo è una buona idea provare fin dall'inizio ad usare la tecnica professionale della scrittura a cinque dita che battere con un dito solo: nei lunghi lavori ciò fa risparmiare moltissimo tempo. Qualsiasi cosa digitiate sulla tastiera, se sono state effettuate le corrette connesioni con il video, apparirà automaticamente sullo schermo.

L'Apple II ha numerose istruzioni chiamate «*istruzioni editoriali*». Queste vi rendono particolarmente semplice scrivere, cioè correggere errori, effettuare modifiche e far apparire sullo schermo il testo corretto. Naturalmente ciò è diverso dai pacchetti specializzati di trattamento dei testi che rendono l'Apple II una macchina da scrivere per ufficio ad alta efficienza. I pacchetti di istruzioni editoriali più avanzati dell'Apple II sono stati progettati con molta cura. Con un po' di pratica li potrete usare facilmente.

L'Apple II vi permette di visualizzare oltre a lettere e numeri anche semplici disegni. Questa possibilità è chiamata «*grafica*» ed è un importante modo di usare l'elaboratore. L'uso di disegni, diagrammi e grafici rende più agevole la presentazione delle informazioni e può essere usato nei giochi, nelle applicazioni gestionali e nei programmi educativi.

L'Apple II ha una grafica eccellente e di facile uso, e potrete imparare a fare semplici disegni seguendo le istruzioni di questo libro. Se desiderate comporre sofisticati e dettagliati grafici potete acquistare una particolare scheda.

L'Apple II è agile e compatto ed è piccolo a sufficienza per essere spostato da stanza a stanza, da casa a casa ed anche da classe a classe. Può essere istallato nella sua nuova collocazione rapidamente e facilmente poiché basta soltanto collegarlo alla corrente elettrica. Inoltre, appena acceso è pronto a ricevere, digitati sulla tastiera, comandi che debbono solo essere scritti in un linguaggio compreso dall'elaboratore. Il linguaggio è il BASIC che vi permette di dare comandi che sono rapidamente ed automaticamente eseguiti dall'Apple.

Come si è sviluppato l'Apple II?

I due principali eventi che hanno accelerato l'avvento della microelettronica ed in particolare dei microprocessori sono la gara spaziale degli anni sessanta e la necessità, fin dalla fine della seconda guerra modiale, delle varie organizzazioni di difesa del mondo, in particolare il Dipartimento della Difesa degli USA e del Ministero della Difesa inglese. Nella gara spaziale, poiché i missili americani avevano minor potenza di quelli russi, gli americani hanno avuto la necessità di ridurre dimensioni e peso delle apparecchiature dei propri missili comprese quelle elettroniche. Nello stesso tempo i capi della difesa cominciarono a richiedere elaboratori che fossero piccoli, compatti ed adatti ad essere trasportati in battaglia, senza danno. In particolare stimolarono le industrie elettroniche americane a ricercare e sviluppare sistemi di miniaturizzazione dei circuiti elettronici. Risultato di ciò è il microprocessore spesso chiamato *chip*. Questo non soltanto

è estremamente piccolo (più piccolo di un'unghia) e relativamente potente (della stessa potenza dei primi elaboratori che occupavano un'intera stanza), ma è anche un'apparecchiatura multifunzionale che può svolgere ogni funzione elettronica per la quale sia stato programmato. Questa versatilità ha portato all'uso del microprocessore in un ampio e sempre crescente campo di applicazioni (la più comune e più conosciuta è il «cuore» di un elaboratore multifunzionale come l'Apple II). La conseguente produzione di massa ha permesso la diminuzione del costo (del microprocessore, non del microelaboratore) per unità, a poche lire.

L'Apple II deve il proprio sviluppo all'intuito di due giovani americani. La storia dell'Apple cominciò nel 1976 quando due tecnici autodidatti collaborarono a costruire una piccola scheda di elaboratore per uso personale. Steven P. Jobs, 21 anni, e Stephen G. Wozniak, 26 anni impiegarono sei mesi per progettare il proprio prototipo, ma solo quaranta ore per costruirlo. Ebbero immediatamente un ordine per 50 elaboratori.

Con questo primo ordine in mano, si procurarono circa 1350 dollari vendendo un pulmino Volswagen usato ed una calcolatrice programmabile ed allestirono un laboratorio nel garage di Jobs. Meno di un anno dopo, alla fine del 1976, furono in grado di fondare la Apple Computer Company, con Jobs come direttore commerciale e Wozniak come tecnico. Chiamarono «Apple» l'elaboratore e la società forse perché la mela (apple) rappresenta la semplicità che si erano sforzati di raggiungere nel progetto e nell'uso del proprio elaboratore.

Il primo elaboratore, venduto in kit ad appassionati di elettronica, fu un tale successo che la richiesta rapidamente superò sia il capitale sia la potenzialità. Ritenendo che questo fosse un prodotto commerciale, Jobs e Wozniak trovarono allora un dirigente che potesse aiutarli a rendere la società finanziariamente solida. Il primo assunto fu A. C. Mike Markkula, conosciuto attraverso un amico comune. Markkula era stato già con pieno successo responsabile del marketing di due società di semiconduttori con esperienza di

sviluppo dinamico: Intel Corporation and Fairchild Semiconductor.

Dopo aver individuato il mercato del nuovo elaboratore ed aver valutato le possibilità della Società Apple, i tre decisero l'assetto del capitale, la dirigenza, le innovazioni tecnologiche, lo sviluppo del software ed i marketing. Il finanziamento iniziale della Apple venne da Markkula e da un gruppo di capitalisti. Da questo piccolo avvio la compagnia dilagò. Ora impiega più di duemilacinquecento persone negli uffici, nelle fabbriche, nella distribuzione e nel supporto in tutto il mondo. L'Apple rimase una società privata fino al dicembre 1980, quando si fece un'offerta pubblica di 4,5 milioni di azioni ordinarie. Nel maggio 1981 ci fu una seconda offerta di 2,3 milioni di azioni ordinarie vendute a circa cento azionisti che acquistarono le azioni secondo un piano di investimento o privatamente.

L'Apple è passata in quattro brevi anni da laboratorio di due sole persone in un garage, a società internazionale che nel 1981 ha venduto per più di 34 milioni di dollari.

Oggi l'Apple Computer Inc. progetta, fabbrica e vende i suoi Personal ed i sistemi per applicazioni nel campo dell'educazione, degli affari, della scienza e per la casa. Le fabbriche dell'Apple si trovano a Cupertino (California), Carrolton (Texas), a Cork, in Irlanda e a Singapore. Inoltre ci sono anche sette centri di supporto regionali in America ed in Europa ed i prodotti Apple sono venduti in tutto il mondo attraverso una rete di più di duemilacinquecento rivenditori. Fondata nel 1977, l'Apple è una delle maggiori fabbriche di Personal. La maggior parte degli impiegati ha una laurea e più della metà di questo personale è impiegato nello sviluppo del software.

I diversi prodotti sono l'Apple II ed il sistema Apple III (nel gennaio 1983 è uscita una nuova versione, l'Apple IIe).

Inoltre l'Apple progetta e produce i propri drive e sviluppa la maggior parte del software applicativo per i propri elaboratori. A causa dell'ampia diffusione, molte società indipendenti producono accessori e scrivono programmi per l'Apple. Ciò vi assicura un'ampia scelta di hardware e software per espandere il vostro sistema.

L'Apple II è stato il primo Personal completamente programmabile. Si colloca nella linea dei prodotti a basso costo come Personal facile da usarsi per piccole imprese, scuole e tecnici. È tuttavia un po' più caro della nuova categoria di elaboratori domestici (home computer) che ora sono in produzione.

Il sistema Apple II standard ha fino a 128K di memoria ROM e fino a 64K di memoria RAM. Il sistema, molto compatto, contiene anche la tastiera, l'interfaccia per il registratore a cassetta, il connettore I/0 per i giochi e fino ad otto connettori per periferiche.

È stato progettato per una facile utilizzazione sia da parte del principiante sia da parte di persone con esperienza sugli elaboratori. L'Apple è stato il primo Personal con il BASIC inserito nel sistema. È inclusa la grafica ad alta risoluzione di colore, il sistema operativo a dischi e molti linguaggi di programmazione. Da quando è entrato in produzione, sono stati sviluppati moltissimi programmi applicativi che ampliano le possibilità di applicazione.

L'Apple usa il microprocessore 6502, simile a quello utilizzato nei primi giochi elettronici ed identico a quello usato da molti successivi microelaboratori. Ad esempio il CBM Pet. Per ironia questo microprocessore trae origine dalla microelettronica militare ed è usato nel sistema di guida dei missili balistici intercontinentali ed anche nei modernissimi missili tipo Exocet. Questo microprocessore è il cuore (ed il cervello) dell'Apple. Benché esso faccia un duro lavoro, tutte le elaborazioni ed i calcoli, voi potete utilizzare il suo potenziale e farlo lavorare per voi senza sapere come funzioni.

Le dimensioni di un elaboratore, ed in molti versi la sua potenza, sono date dalle dimensioni di memoria, soprattutto della RAM. È misurata in «K bytes». K è l'abbreviazione per chilo o mille (è in effetti eguale a 1024). Un byte è formato da otto bit, ma ciò che importa ricordare è che serve circa un byte per memorizzare un solo carattere. Così un Apple II con 32K può memorizzare poco più di 32.000 caratteri nella propria memoria centrale.

Quando usate l'elaboratore, le memorie RAM e ROM memorizzano sia i programmi che devono funzionare sia il

linguaggio interprete che converte le istruzioni che avete introdotto. Così molti dei nuovi piccoli elaboratori domestici come il PET ad 8K, lo ZX80 ad 1K fino allo Spectrum di 16K, possono utilizzare semplici programmi di giochi ma non sono probabilmente grandi abbastanza per utilizzare sofisticati giochi di fantasia quali i Dungeons ed i Dragons; e sicuramente non sono grandi a sufficienza per essere usati per i programmi gestionali più ampi.

Cosa può fare l'Apple?

L'Apple II può fare ogni cosa voi gli diciate di fare. Cioè obbedisce ad ogni istruzione od insieme di istruzioni che siano date correttamente. Un'insieme di istruzioni per un elaboratore è chiamato programma ed è scritto in una lingua speciale chiamata «linguaggio di programmazione». Come ogni altro elaboratore l'Apple esegue programmi e fa ciò che gli dite di fare. Così l'unico modo per usare l'elaboratore è imparare a programmare nella sua lingua che è il BASIC. Benché il BASIC sia la lingua naturale dell'Apple, non è la lingua naturale del microprocessore 6502. Il programma in BASIC deve essere tradotto nel linguaggio o codice (detto linguaggio macchina) compreso dal microprocessore. Questo secondo livello di traduzione è svolto automaticamente quando fate «*girare*» un programma e non ve ne accorgerete quando userete l'elaboratore. È possibile scrivere un programma direttamente nel linguaggio macchina dell'Apple ma questo libro non tratterà questo argomento. Scrivere i programmi in linguaggio macchina è notevolmente più difficile dello scriverli in BASIC anche se essi permettono di operare ad una velocità maggiore. La ragione dell'aumento di velocità è che si perde tempo nella traduzione dal BASIC al linguaggio macchina.

Un altro motivo è che l'originale è sempre migliore della traduzione. Come una traduzione parola per parola di Shakespeare in francese non può essere migliore dello Shakespeare in inglese, o di Molière in francese, così una traduzione dal Basic in linguaggio macchina non può essere più effi-

ciente di un programma scritto direttamente nel linguaggio macchina.

Tuttavia non è indispensabile essere un esperto programmatore per usare l'Apple II poiché si può acquistare un sempre maggior numero di programmi pronti. Bisogna tener presente che questi comprendono anche una quantità di altri linguaggi di programmazione e perfino differenti versioni del BASIC. Questi programmi si trovano sia su nastro, dal quale vengono trasferiti nella memoria per mezzo di un normale registratore; oppure, meglio, su un dischetto usato con un drive. Poiché sono già disponibili molti programmi, potreste avere interesse a dare un'occhiata agli annunci pubblicitari su riviste di elaboratori per giochi od affari. Spesso gli stessi programmi sono indicati come «software», al contrario dello stesso elaboratore che è noto come «hardware».

È stata l'evoluzione del software di facile uso che ha reso pratici e famosi gli elaboratori come l'Apple II. Questi programmi di software, utilizzabili su appositi dischi (vedi fig. 1.3), forniscono le istruzioni che dicono all'elaboratore che cosa fare e come. In effetti essi configurano il sistema per svolgere i compiti assegnati.

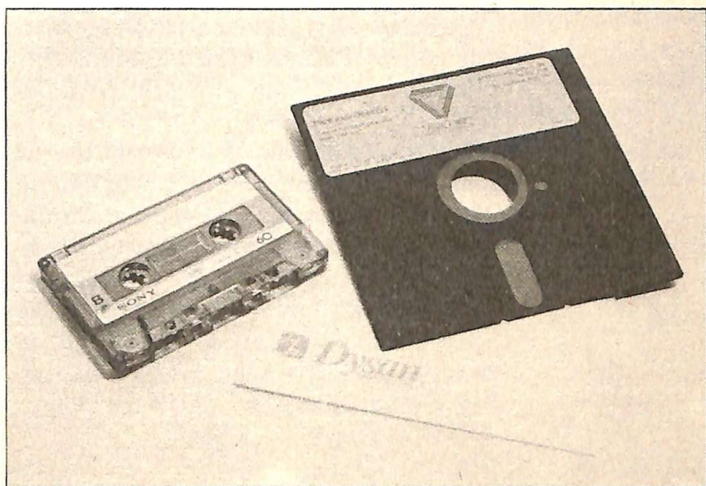


Fig. 1.3 Una cassetta ed un dischetto

L'Apple offre una delle più vaste scelte di applicazioni e programmi rispetto a qualsiasi produttore di Personal. La validità di un pratico software è una delle cause più importanti della popolarità degli Apple nelle applicazioni sofisticate. Queste comprendono programmi scritti da programmatori Apple e software scritto da utenti dell'Apple, il quale è poi normalmente perfezionato e corretto da personale Apple, prima della vendita.

I pacchetti software sono raccolti in due categorie:

— Software di applicazione, che trasforma l'elaboratore in una macchina specializzata in compiti specifici, come contabilità, scrittura di testi, memorizzazione di elenchi ed indirizzi;

— Libreria Linguaggi che comprende programmi di sviluppo che permettono agli utenti più sofisticati di ottenere il massimo vantaggio dal loro Apple per mezzo di linguaggi avanzati.

Alcuni esempi di applicazioni di software:

Il Controller:	un piccolo sistema per la gestione d'affari e la contabilità. Consiste in conti attivi, passivi e libri mastri; questo programma offre all'uomo d'affari una facile, reale alternativa alla contabilità manuale.
L'Apple Cashier:	migliora il controllo dell'inventario e la raccolta di documenti permettendo al piccolo proprietario d'azienda di formare e mantenere archivi di clienti e venditori e di far scrivere fatture e relazioni.
L'Apple Writer:	trasforma un Apple 2 in un sofisticato redattore di testi, che redige e stampa in modo rapido ed a buon mercato un'ampia varietà di documenti e di affari, scientifici e legali.
L'Apple Plot:	permette all'utente di creare, correggere e stampare diagrammi e grafici particolarmente dettagliati.

L'Apple Post: un sistema di gestione della lista di indirizzi che permette all'utente di inserire, redigere ed immagazzinare nomi, indirizzi e numeri telefonici.

Oltre a questi ed ad altri programmi applicativi, l'Apple offre una linea completa di linguaggi di programmazione. Essi comprendono il BASIC, il PASCAL, il FORTRAN, il PILOT. Tutti e quattro sono ampliamente usati nell'istruzione; ci sono già vaste librerie-programmi che usano questi linguaggi.

L'Apple II è in grado di fare molte cose. Come con la maggior parte dei microelaboratori, potete fargliele svolgere senza avere alcuna conoscenza di programmazione. Tuttavia è spesso utile saper programmare, od almeno correggere o modificare un programma preesistente. Inoltre programmare è divertente. È facile e vi fornisce il mezzo per esprimere e comunicare le vostre idee all'elaboratore che poi le prova per voi.

Come si amplia l'Apple

Oltre a svolgere calcoli ed immagazzinare informazioni, l'Apple II può, molto più di altri elaboratori essere usato in collegamento con altri dispositivi. Le unità che si possono collegare ad esso, sotto il suo controllo, si dicono «periferiche». Ne avete incontrata una nella parte precedente: il drive. Essa è un'unità periferica usata per memorizzare in modo permanente le informazioni od i programmi per mezzo di

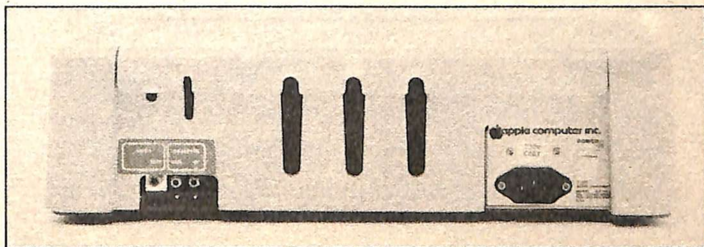


Fig. 1.4 La parte posteriore dell'Apple II

piste magnetiche sulla superficie di un dischetto. L'Apple II ha un cavo che lo collega al drive. La fig. 1.4 mostra il retro dell'elaboratore con i vari connettori per le periferiche.

Per molte applicazioni dell'elaboratore è utile avere i risultati del lavoro in forma scritta, per poter registrare in modo permanente i risultati dei calcoli. È anche di utile aiuto, quando incominciate a scrivere il programma, ed in special modo quando cominciate a fare correzioni, avere una copia scritta. L'output scritto è detto «stampata» o «listato». Il termine stampata si riferisce normalmente ai risultati di una elaborazione, mentre il listato si riferisce alla copia stampata di un programma. Naturalmente una stampante è una periferica indispensabile per ottenere stampate e listati. Può essere attaccata all'Apple come mostra la fig. 1.5. L'Apple produce molte stampanti che potete usare, secondo le vostre precise necessità. Si può anche usare una stampante prodotta da altre case.

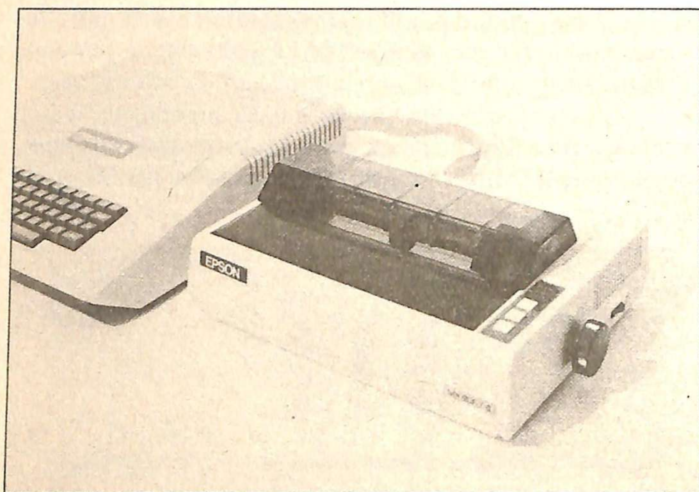


Fig. 1.5 Una stampante collegata all'Apple II

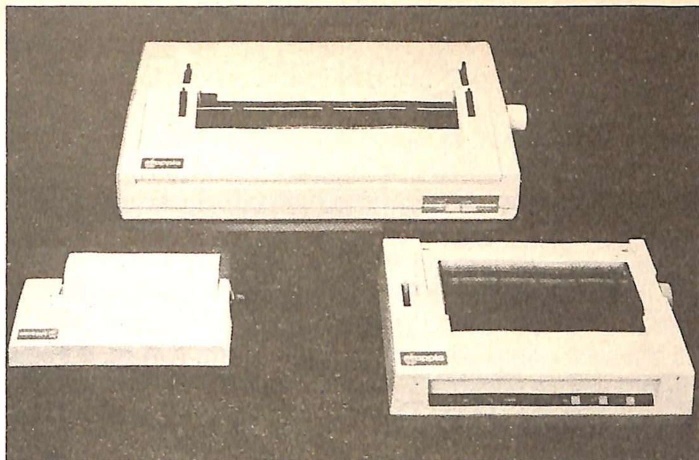


Fig. 1.6 Alcune stampanti per l'Apple

L'Apple ha annunciato, nel gennaio 1983, delle nuove stampanti che si aggiungono alle stampanti tecniche Silentype (vedi fig. 1.6) e precisamente la stampante a punti e la stampante di qualità. La stampante a punti stampa caratteri formati da un'insieme di punti. La definizione di tali caratteri dipende dal numero di punti usati e da quanto siano posti vicino l'uno all'altro. La stampante Apple produce output grafici con una matrice di 7×9 punti per un totale di 144×160 punti per pollice quadrato (più di ventitremila). Può stampare in entrambe le direzioni ad una velocità di 120 caratteri al secondo. Usa una interfaccia parallela.

La stampante di qualità è una stampante a margherita bidirezionale che può stampare sino a 40 caratteri al secondo. Usa una interfaccia seriale.

Tutte le periferiche connesse all'Apple II utilizzano i connettori (porte) posti nel lato posteriore dell'elaboratore. «Porta» è un altro modo di chiamare il punto di connessione tra un elaboratore ed il mondo esterno poiché è simile al

modo in cui un porto od un aereoporto collega un paese od una regione con il resto del mondo. Fin dal principio l'Apple è stato progettato per svilupparsi insieme alle necessità dell'utente. Ciò è sfociato in un sistema base che comunica facilmente con altri elaboratori e periferiche ed in una serie di accessori e periferiche progettate dall'Apple. C'è anche una vasta gamma di accessori progettati e costruiti da altre società.

Basta un solo esempio per spiegare questo punto. Il sistema grafico Robocom BIT STIK è il primo di una nuova generazione di elaboratori grafici che sono in grado di eliminare il tempo necessario ad introdurre i dati. Ciò dà il via ad infinite nuove possibilità per elaboratori grafici economici, che permettono agli utenti di microelaboratori di fare grafici multicolori, disegni tecnici, ecc. con facilità e velocemente e con la minima conoscenza di elaboratori.

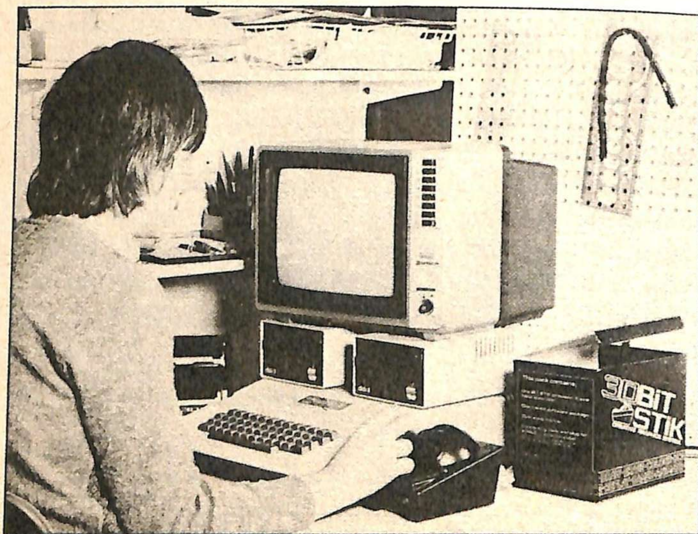


Fig. 1.7 Il sistema grafico BIT STIK

Una serie di cartoline di interfaccia permette all'Apple di comunicare con gli altri elaboratori, stampanti, driver, terminali video ed altre attrezzature di elaboratori. Questa disponibilità di un'ampia gamma di periferiche e di software permette una flessibilità di sviluppo per tutti gli Apple.

Quali sono le applicazioni dell'Apple II?

L'Apple II è stato progettato per molte applicazioni serie. In generale le aree in cui può essere usato sono: nel commercio e nell'industria, per uso personale e di ufficio; nelle scuole e, fino ad un certo livello, in casa.

Per l'uso domestico vi sono programmi già pronti e giochi di ogni genere ed altri sono disponibili continuamente. Ve n'è uno, alla fine del libro, chiamato Torre di Hanoi (un classico ed antico problema di logica) che voi potete inserire nell'Apple per giocare. Spesso si critica che uno strumento elettronico avanzato come l'elaboratore sia usato per uno scopo frivolo come i giochi e non vi è dubbio che molti giochi siano frivoli. Tuttavia vi sono giochi fantasiosi e stimolanti come la Torre di Hanoi, che hanno un ben preciso valore educativo. Altri giochi possono insegnare od aiutare lo sviluppo di capacità che vanno dalla semplice capacità di manipolazione e di coordinamento nei bambini, alle logiche mentali richieste per trovare soluzioni ai puzzles, per provare strategie e tattiche contro le situazioni presentate dall'elaboratore. Ad esempio vi sono dei programmi sul gioco degli scacchi veramente formidabili ed in breve tempo potrete scrivere voi stessi i vostri programmi di giochi.

La presenza in casa dell'Apple II permette di non restringere le attività educative alla sola scuola. L'elaboratore dà la possibilità di avere sempre a disposizione i programmi educativi e l'Apple II ha la più grande libreria (fig. 1.8) di programmi educativi disponibile per un elaboratore. La si può effettivamente usare sia a casa sia a scuola. L'elaboratore che aiuta l'apprendimento, inoltre, non vuole sostituirsi all'insegnante, ma aiutarlo fornendogli un altro mezzo. In una società post industriale (è spesso così chiamata; come siamo passati attraverso una rivoluzione industriale nell'al-

tro secolo, così ora stiamo attraversando la rivoluzione della tecnologia e dell'informazione) è importante far conoscere, nel modo più veloce possibile, la tecnologia attuale. Solo così i giovani possono conoscere la potenzialità dell'elettronica moderna ed essere in grado di sfruttare a fondo la potenzialità dell'elaboratore.

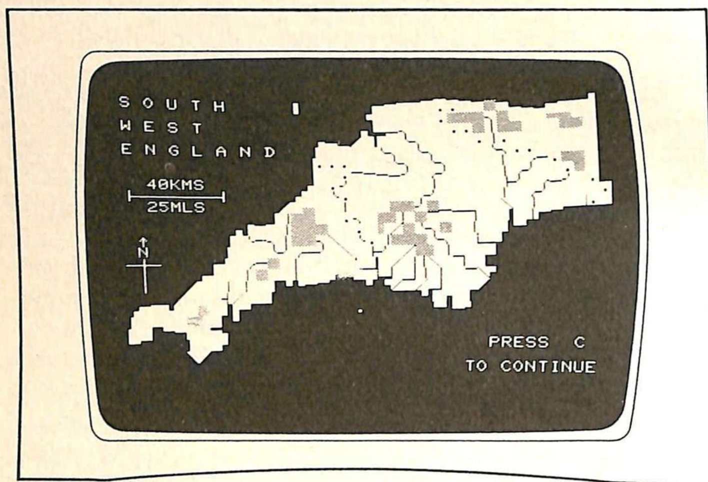


Fig. 1.8 Immagine prodotta dal programma «Geography Taster»

La presenza quotidiana dell'Apple II a casa od a scuola è un mezzo che ci può aiutare a raggiungere tale obiettivo. Il microelaboratore ha nella scuola molte valide utilizzazioni che vanno dall'aiuto all'apprendimento, dai programmi di istruzione ai quiz. Nell'insegnamento superiore l'Apple è ideale per progetti di apprendimento programmato.

Il principale uso dell'Apple II è nel campo gestionale. Quasi sempre, ma non sempre, le applicazioni gestionali richiedono la velocità e la capacità dei drive per dischetti. Per

tutte le attività in cui è necessario memorizzare una grande quantità di dati e rileggerne alcuni (ad esempio esaminare il livello di giacenza di determinati oggetti) è indispensabile usare dischetti e drive per memorizzarli. Ciò serve non solo perché il dischetto ha una grande capacità di immagazzinare dati, ma soprattutto perché permette di rileggere od «accedere ad» un'informazione in pochissimo tempo. Una informazione memorizzata in un dischetto può essere riletta fino al punto richiesto e, per questo motivo, una unità a disco a volte è chiamata ad «accesso casuale», od ad «accesso diretto». Al contrario, usando un registratore è necessario far svolgere il nastro fino al punto desiderato. Ciò naturalmente significa inutili e lunghi ritardi ed è particolarmente irritante quando sono necessari numerosi accessi all'informazione.

Forse il più famoso e più venduto software gestionale per l'Apple II è il VisiCalc, un programma che permette all'utente di elaborare modelli gestionali in un tempo infinitamente più breve di quello che altrimenti ci vorrebbe.

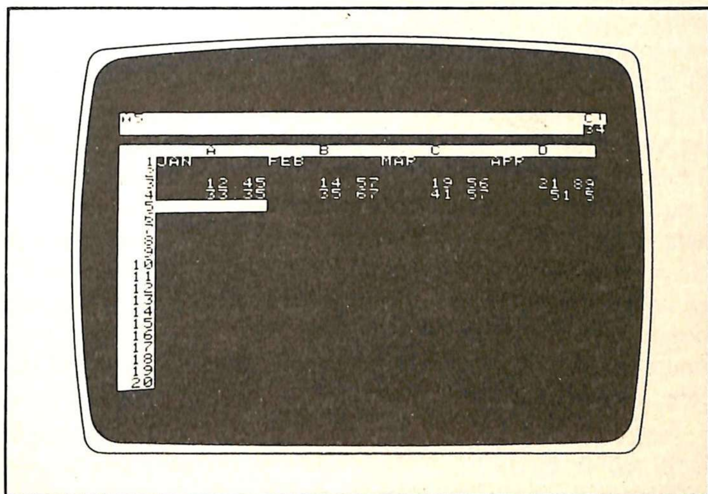


Fig. 1.9 Cosa appare sul video con il «VisiCalc»

Il programma trasforma lo schermo in un taccuino su cui vengono definite formule matematiche e relazioni, l'elaboratore fornisce i risultati. Il programma è chiamato dinamico perché potete cambiare le parti all'istante. Ciò vi permette di eseguire prove del tipo «che succederebbe se?».

Ciò vuol dire che potete cambiare un determinato valore nel modello e immediatamente l'elaboratore calcolerà i risultati di tale modifica nella parte rimanente del modello.

Per esempio, potreste cambiare il numero degli impiegati e l'elaboratore immediatamente vi mostrerà gli effetti della modifica sulle vostre uscite di cassa.

Una seconda applicazione dell'Apple II è il trattamento dei testi.

Il trattamento dei testi è l'applicazione della tecnologia dell'elaboratore all'automatica produzione di documenti. Ciò vuol dire che lettere, relazioni e in effetti tutto ciò che debba essere stampato, può essere fatto per mezzo dell'elaboratore.

Questo libro è stato scritto con un sistema di trattamento dei testi. Per uso commerciale un trattamento di testi ha bisogno di una stampante di qualità (meglio una stampante a margherita per evitare che l'output sia scadente). Ci sono molti programmi di trattamento di testi che usano l'Apple II. Lo Zardax è un pacchetto australiano. Richiede l'Apple II Plus. È sofisticato, tuttavia di facile uso. Con esso potete scrivere sullo schermo ogni testo di lunghezza superiore ai 13500 caratteri e anche due volte tale lunghezza se ampliate l'hardware dell'Apple. Lettere o altri documenti possono essere stampati da quasi tutte le stampanti. Lo Zardax lavora sia su documenti lunghi sia su corti. I più corti possono essere collegati tra loro per formare lunghe relazioni. Oppure potete formare sul dischetto un file con, per esempio, 100 nomi ed indirizzi e poi usare lo Zardax per produrre 100 lettere individualizzate.

Inoltre, poiché il sistema usa files standard dell'Apple DOS, potete scrivere i vostri programmi per modificare questi files di indirizzi: per esempio classificare, esaminare e selezionare.

Un altro importante programma di trattamento di testi è il Format 80. Per usare il Format 80 dovete avere un Apple II od un Apple II Plus con un Applesoft in ROM ed almeno 48K di RAM; almeno un drive ed una cartolina di controllo dell'unità disco, un video a 80 colonne e una stampante con l'interfaccia adatta.

Le attuali capacità del Format 80 sono simili a quelle dello Zardax. Tutte queste attività e molte altre possono essere eseguite usando programmi esistenti. Tuttavia se imparate a programmare con l'Apple II, potrete scrivere da soli i vostri programmi (vale la pena di ricordare che un certo numero di programmi commerciali di gran successo sono stati scritti perché la gente comune a casa, a scuola o in ufficio non era soddisfatta dei programmi acquistati, cosicché li ha scritti da sé).

Non soltanto potete esprimere le vostre idee, ma potete anche adattare alle vostre esigenze i programmi che acquistate.

Poiché l'Apple è molto più veloce nei calcoli delle persone comuni, potreste considerare pratico affidare all'Apple tutti i vostri calcoli complessi. Inoltre l'Apple può anche servire, in questa nuova era dell'informazione, per estendere ed ampliare la capacità intellettiva dell'uomo, se usato come fonte di intrattenimento e come mezzo di educazione e di affari.

Sommario

L'Apple II è un piccolo microelaboratore che può avere una memoria fino a 64K. La parola «micro» è usata per descrivere le dimensioni fisiche della macchina ed in particolare del processore e non la sua abilità nel calcolo. Tuttavia sono le piccole dimensioni di un microprocessore che lo rendono di grande potenza per essere usato come uno strumento personale a casa, a scuola, in ufficio. Le piccole dimensioni sono la diretta conseguenza di recenti sviluppi tecnologici, stimolati soprattutto dalla rivalità tra le maggiori nazioni del mondo.

L'Apple può essere usato in moltissimi casi, ma le sue specifiche aree di applicazione sono l'ufficio e l'educazione; inoltre, in minor misura, si sta estendendo l'uso domestico.

Si possono già comprare programmi per svolgere molti incarichi in questi campi. Ciò permette all'Apple di essere vantaggiosamente usato subito dopo l'acquisto, senza alcuna esperienza di programmazione. Ovviamente con il passare del tempo, moltissimi programmi diventeranno accessibili. Le possibilità dell'Apple possono essere ampliate in moltissimi modi, con l'acquisto di parti supplementari o periferiche, come una stampante che può essere collegata all'elaboratore e usata con esso.

Capitolo 2

L'uso dell'Apple II

Come abbiamo spiegato nel primo capitolo, l'Apple II è davvero un sistema di elaborazione piuttosto che un semplice elaboratore.

Ci sono così tanti accessori e periferiche disponibili per l'Apple II, che è veramente possibile (ma non probabile) che possiate avere un sistema Apple II unico.

Per questo motivo dobbiamo definire molto attentamente il sistema Apple II usato per gli esempi di questo libro.

Il sistema Apple II usato per questo libro (v. fig. 2.1) è un

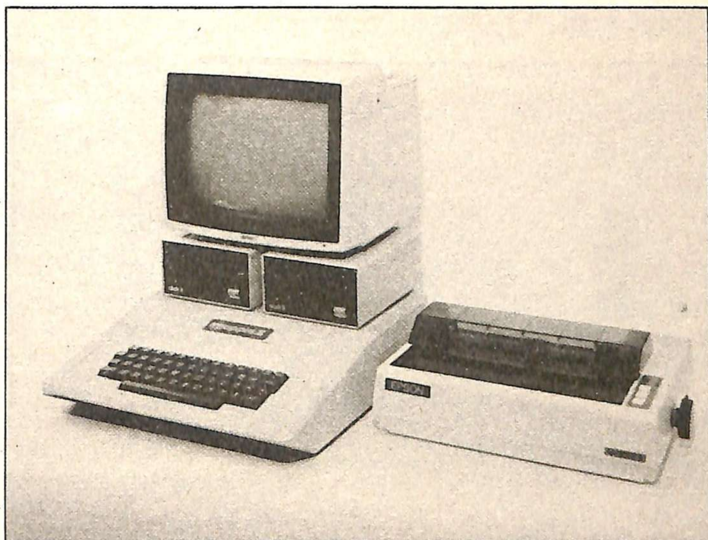


Fig. 2.1 Il sistema Apple II

Apple II Europlus con 48K di RAM (è una delle più comuni configurazioni in uso in Europa) con 2 drive Apple, un video BMC ed una stampante a matrice Epson MX 80.

Accensione

Diversamente da molti nuovi elaboratori domestici, l'Apple II può essere direttamente collegato alla corrente perché c'è un trasformatore all'interno dell'Apple II che fornisce i 4 voltaggi 5 V.; —5,2 V.; 11,8 V.; —12 V. di cui la parte elettronica interna dell'Apple II ha bisogno.

Un filo elettrico può essere collegato direttamente nella presa situata sul lato destro del pannello posteriore dell'Apple II. Vicino alla presa c'è l'interruttore.

Prima di accendere l'Apple II, assicuratevi di aver correttamente collegato tra loro le tre fondamentali e necessarie parti del sistema e controllate che sia pronto un dischetto. Insieme ci sono il video, l'elaboratore vero e proprio, almeno un drive ed un dischetto. Se il vostro sistema non ha un drive, potrà egualmente essere usato con un registratore a

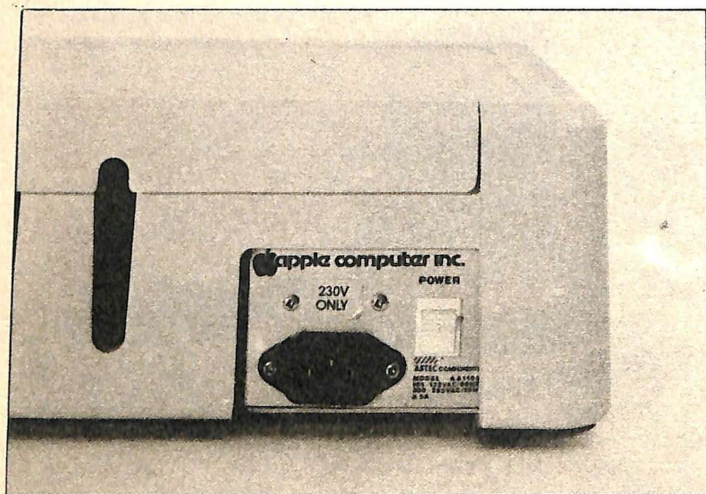


Fig. 2.2 Vista posteriore dell'Apple II con l'interruttore d'accensione

cassetta e la cassetta standard è il mezzo per registrare e memorizzare. Il video presenta un'immagine grafica di ciò che digitate sulla tastiera. L'elaboratore legge i caratteri che digitate ed obbedisce alle istruzioni o semplicemente le stampa sul video.

Il dischetto assomiglia molto ad un piccolo e sottile disco a 45 giri contenuto in una custodia di cartone.

È attualmente fatto in plastica flessibile con un rivestimento di ossido magnetico. Memorizzate sulla superficie del disco ci sono delle istruzioni speciali che fanno funzionare l'elaboratore. Naturalmente potete anche usare un solo dischetto per memorizzare i vostri programmi e le informazioni.

Il dischetto (fig. 2.3) è il mezzo usato per prendere l'informazione dall'elaboratore ed immagazzinarla (scrittura) nel disco.

Il modo in cui ciò avviene è molto simile al modo in cui un registratore può memorizzare e trasmettere la musica. Per cominciare, togliete il dischetto dalla busta di carta, facendo molta attenzione a non toccare qualcuna delle parti allo scoperto del disco stesso. Se prendete soltanto il disco dall'eti-



Fig. 2.3 Un drive

chetta, (fig. 2.4) potrete stare tranquilli. Alzate lo sportelletto al centro della parte anteriore del drive. Prendendo con delicatezza il disco con il pollice sull'etichetta, inseritelo lentamente nella fessura del drive, fino a che non sia infilato completamente. Non piegate o forzate il disco. Se non entra facilmente, toglietelo e riprovate. Premete lo sportelletto fino alla chiusura. Il vostro Apple II è pronto. Accendete il video ed infine spingete l'interruttore posto nella parte posteriore dell'Apple. L'Apple, essendo stato correttamente collegato alla presa corrente, funzionerà (alcuni preferiscono accendere il sistema prima di infilare i dischetti. Entrambi i metodi vanno bene).

L'Apple emette un «bip». Si accenderà sia una piccola luce rossa sul davanti del drive sia il tasto cõn la scritta POWER sulla sinistra in basso della tastiera. La luce bianca rimarrà accesa finché lo sarà l'elaboratore. La luce rossa rimarrà accesa finché sarà usato il drive. In questa fase resterà accesa solo alcuni secondi e potrete sentire che il drive ronza appena. Questo è il drive che legge il sistema operativo dal disco e lo trasferisce nell'elaboratore. Dopo pochi istanti il ronzio termina e la luce si spegne. L'Apple II è pronto.

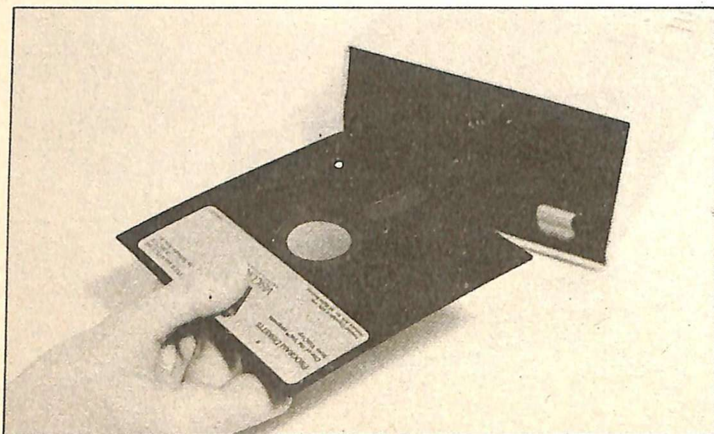


Fig. 2.4 Inserimento di un dischetto

Il video

In basso a sinistra verso l'angolo, nel pannello posteriore dell'Apple II, vi è un connettore metallico contrassegnato con VIDEO. Questo connettore vi permette di inserire un cavo tra l'Apple II ed il video. Se la connessione è fatta correttamente il video mostrerà quello che voi digitate sulla tastiera. (Dovrete comprare uno speciale accessorio per convertire il segnale prodotto dall'Apple II in quello necessario per un televisore con standard europeo. Naturalmente sempre che voi usiate il televisore come video).

Lettere, simboli e numeri che voi digitate sulla tastiera sono detti «caratteri». Il video è largo tanto da contenere 40 caratteri su una sola riga. Se la linea avesse più di 40 caratteri, il cursore si sposterà automaticamente all'inizio della riga successiva.

Il video contiene 24 linee ognuna di 40 caratteri. Un carattere può quindi essere posto in ognuna delle 960 (24×40) posizioni dello schermo.

Quando l'ultima riga dello schermo è piena, il contenuto dello schermo automaticamente si sposta in alto di una riga lasciando bianca l'ultima riga. Questo è chiamato «scorrimento verso l'alto». La precedente riga superiore scompare.

La tastiera

La tastiera è dello stesso tipo di una ordinaria tastiera di macchina da scrivere. È conosciuta come tastiera «QWERTY» perché così è organizzata la prima riga di lettere letta da sinistra a destra. I tasti formano numeri, lettere e simboli. Vi è una barra spaziatrice che introduce spazi tra le parole. La tastiera si usa come quella di una normale macchina da scriver. Vi sono inoltre un certo numero di tasti speciali che illustreremo in questo paragrafo.

Cominciamo con quello che è un «non tasto» <SHIFT>. Tutti i caratteri che voi digitate sulla tastiera sono visualizzati maiuscoli sullo schermo. Il tasto <SHIFT> è usato per scrivere i simboli speciali che appaiono sulla parte alta dei tasti. Spinto da solo, il tasto <SHIFT> non visualizza nulla. Spinto con un altro tasto,

visualizza il simbolo speciale. per esempio, spingendo <SHIFT> 1 visualizzerà il «!».

L'Apple II può soltanto riconoscere le maiuscole. La tastiera ha anche il tasto «1» (uno) ed il tasto «0» (zero) nella prima riga. Nelle normali macchine da scrivere sono spesso scritti con il carattere minuscolo «l» (per uno) ed il carattere maiuscolo «O» (per zero). Tutti gli elaboratori richiedono, inoltre, che si faccia molta attenzione con le lettere ed i numeri. I numeri sono trattati dal sistema in modo completamente diverso dalle lettere e non possono essere scambiati. Per questo motivo troverete spesso lo zero scritto come «∅.» per distinguerlo dalla lettera «O».

Ci sono 6 tasti speciali a cui bisogna fare particolarmente attenzione; sono: <RETURN>, <RESET>, <CONTROL>, <ESCAPE>, <LEFT ARROW> (←) e <RIGHT ARROW> (→); il tasto <RETURN> è sul lato destro della tastiera. Spesso è simile al tasto di «ritorno carrello» di una macchina da scrivere, poiché spesso sposta il cursore (il rettangolino lampeggiante che vi indica la posizione) all'inizio della nuova riga. Nell'Apple II è anche usato per dire all'elaboratore che siete soddisfatti di quel che avete digitato e che volete che l'elaboratore lo usi. In altre parole, spingendo il tasto <RETURN> introducete ciò che avete digitato nell'elaboratore e chiedete all'Apple di darvi la risposta. Per esempio, se avete digitato un'istruzione premendo il tasto <RETURN> dite all'elaboratore di eseguire questa istruzione.

Una regola generale da ricordare quando state ancora imparando ad usare l'Apple è: «Nel dubbio, spingete il tasto <RETURN>».

Il tasto <RESET> è posto nell'angolo in alto a destra della tastiera. Lo si usa quando si vuol fermare l'elaboratore mentre lavora. Si usa spingendo contemporaneamente i tasti <RESET> e <CONTROL> e funziona quando ne lasciate almeno uno; interrompe l'esecuzione del programma. Questo tasto dovrà essere usato solo come estrema risorsa.

Ci sono due modi migliori per interrompere un programma mentre sta girando. Il più semplice è il tasto <ESCAPE>. Si trova a sinistra del carattere «Q». Spingere il tasto

<ESCAPE> è il mezzo più comune per tornare indietro in un punto noto del programma, ma non può funzionare in tutti i programmi. Il secondo metodo usa il tasto <CONTROL> (subito sotto il tasto <ESCAPE>). Il tasto <CONTROL> è usato insieme ad altri tasti per controllare il modo in cui ogni programma lavora. <CONTROL>/C (spesso scritto come «CTRL/C») è il sistema accettato da tutti gli elaboratori per uscire da un programma. Quando usiamo due tasti separati da una linea obliqua indichiamo che dovete spingerli entrambi contemporaneamente.

Fermare un programma potrebbe sembrarvi una cosa strana, in pratica, vedrete, lo farete frequentemente ed in modo particolare quando comincerete a scrivere un vostro programma.

I tasti <LEFT ARROW> e <RIGHT ARROW> si trovano subito sopra il <RETURN>. Li potete usare per correggere qualche errore sulla riga che state digitando al momento. Questo tasto funziona solo mentre scrivete la riga e non dopo che avete spinto il tasto <RETURN>. Ogni volta che spingete il tasto <LEFT ARROW> il cursore si muove indietro di uno spazio (finché raggiunge l'inizio della riga). Così per fare correzioni, potete spingere il tasto <LEFT ARROW> finché il cursore si trovi sull'inizio dell'errore e potete digitare sopra la versione corretta. Se spingete il tasto <RIGHT ARROW>, il cursore si sposta di un carattere a destra, senza cancellare alcun carattere. Ricordate che quando spingete <RETURN>, tutti i caratteri a destra del cursore vengono automaticamente cancellati.

Caricare un programma

Probabilmente il modo più simpatico ed indolore per far pratica dell'Apple e della tastiera, è usarlo per fare giochi che necessitano di risposte digitate sulla tastiera. Un crescente numero di giochi è già pronto. Li potete leggere sia da disco sia da cassetta. Per caricare, senza curarsi del nome, un programma da cassetta, collegate il registratore all'elaboratore ed inserite la casseta. Ad ogni estremità del cavo di connessione vi sono due connettori: uno nero ed uno grigio. In-

serite il connettore nero nell'uscita «MICROFONO» del registratore e l'altro connettore nero nell'uscita dell'elaboratore «CASSETTE OUT». Inserite il connettore grigio nell'uscita «EAR» del registratore e nell'uscita «CASSETTE IN» dell'elaboratore. Vi chiediamo di avere una cassetta con la registrazione di un programma usato in questo libro. Questo programma è un gioco chiamato la Torre di Hanoi che è chiamato semplicemente «Torre» nella cassetta. Per caricare questo programma nell'elaboratore controllate che tutte le connessioni siano state eseguite correttamente. Riavvolgete la cassetta dall'inizio. Dovete richiamare l'Apple BASIC per ricevere il programma. Se avete una versione recente dell'Apple II, con AUTOSTART, il BASIC sarà caricato automaticamente quando lo accendete. Se avete una versione precedente senza «AUTOSTART» sarà necessario dare il comando:

```
<RESET><CTRL>/B<RETURN>
```

(Se il vostro sistema ha un drive probabilmente non userete il registratore; userete il drive in un momento). Qualunque versione voi abbiate, adesso avrete il BASIC inserito, cioè pronto all'uso. Vedrete una parentesi quadra nel margine sinistro ed un rettangolo lampeggiante. La parentesi quadra significa che il sistema è pronto. È detto pronto perché vi comunica che il sistema è pronto a ricevere le istruzioni, in altre parole vi sollecita a dare i comandi. Il rettangolo lampeggiante è il cursore che vi mostra dove apparirà sullo schermo il carattere che voi digiterete sulla tastiera.

Per caricare il programma dalla cassetta all'elaboratore introducete:

```
LOAD<RETURN>
```

e spingete il tasto PLAY (ascolto) del registratore. L'Apple ora porta una copia del programma dalla cassetta alla memoria RAM. In pratica è molto probabile che non funzioni la prima volta. Quando caricate un programma dal nastro il livello del volume è critico. Se non è correttamente re-

golato apparirà un messaggio del tipo

ERR

oppure

*** MEM FULL ERR

od entrambi

Per trovare la giusta regolazione del volume è necessario usare un metodo empirico. Provate la precedente procedura con il volume regolato basso. Se il comando LOAD ha un buon esito non toccate la regolazione del volume. In caso contrario riavvolgete il nastro, regolate il volume un po' più alto e riprovate. Continuate a fare così finché non funziona. Alla fine la regolazione del livello del volume sarà corretta ed il comando LOAD funzionerà.

Se avete un drive, sicuramente avrete un sistema operativo a dischi ed almeno l'Apple DOS (Sistema Operativo a Dischi) ed il BASIC «Applesoft». In questo caso avrete sullo stesso disco un programma chiamato «COLOR DEMO» o «COLOR DEMOSOFT». (Dovete comprendere che più userete gli elaboratori e più userete parole ed ortografie inglesi).

Il drive è stato probabilmente collegato all'elaboratore dal rivenditore. Se guardate dentro all'elaboratore, vedrete che vi è una cartolina di interfaccia in una delle uscite della scheda-madre. Guardate il glossario per comprendere il significato delle parole che non comprendete. Ci potrà essere confusione perché con il disco avete due versioni del BASIC, oltretutto molto diverse. La prima chiamata INTEGER BASIC e la seconda APPLESOFT BASIC memorizzata nel disco e caricata in memoria dal DOS (le prime versioni avevano in ROM l'INTEGER BASIC, le ultime versioni l'APPLESOFT).

Per usare il dischetto prima accendete l'Apple; inserite il dischetto nel drive, aspettate di sentire uno scatto e chiudete lo sportello. Se avete l'AUTOSTART il dischetto ronzerà

per alcuni secondi ed apparirà sullo schermo il seguente messaggio:

```
DOS VERSION 3.3                         08/25/80
APPLE II PLUS OR ROMCARD           SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CARD)
]
```

Abbiamo già detto che la parentesi quadra significa che il sistema è pronto; cioè è pronto l'Applesoft BASIC, Se non avete l'AUTOSTART digitate

PR#6<RETURN>

Il numero si riferisce all'uscita che ha la cartolina di interfaccia del drive dentro l'Apple. Potrebbe essere diverso. Provate numeri diversi (fino a 7) finché non trovate quello giusto. Se il vostro elaboratore fa qualcosa di strano spingete <CTRL>/<RESET> e riprovate.

Dovreste avere il messaggio «DOS» sullo schermo con una parentesi quadra. Potrete individuare i due differenti BASIC, l'Applesoft e l'Integer dal carattere di pronto che appare sulla sinistra in alto nello schermo. Se è una parentesi quadra è l'Applesoft BASIC; se invece è una parentesi ad angolo è l'Integer BASIC che è così chiamato perché tratta i numeri interi. L'Applesoft BASIC, tra l'altro può trattare i «numeri a virgola mobile» cioè i numeri decimali con virgola. Se volete scegliere i due tipi di BASIC digitate INT per l'Integer e FP per l'Applesoft.

Ora che avete caricato il DOS in memoria avete a disposizione un certo numero di semplici comandi. Digitate uno di questi:

CATALOG<RETURN>

Il disco ronzerà un istante e sullo schermo comparirà una scritta come in figura 2.5.

È un catalogo di tutti i file ed i programmi memorizzati sul dischetto. L'Apple vi mostrerà solo quello che può entrare nello schermo. Se nel dischetto ci fossero ancora altri programmi dovete spingere la barra spaziatrice per vedere il resto.

Fig. 2.5 Il CATALOGO DOS

Capirete di aver visto tutto quando riappariranno il pronto del BASIC ed il cursore lampeggiante.

Osservate il catalogo dei programmi e guardate se ce ne sia uno «chiamato» «Applevision» (Se non ci fosse, ce ne sarà un altro chiamato «DEMO»). Ora digitate:

LOAD APPLEVISION<RETURN>

L'Apple memorizzerà nella memoria centrale il programma chiamato Applevision contenuto nel dischetto. Ora digitate:

RUN<RETURN>

Il programma presente nella memoria centrale sarà eseguito. Applevision (fig. 2.6) è un programma dimostrativo preparato dalla Apple. Vi spiega tre cose (due delle quali capirete meglio dopo aver letto questo libro): grafici (fare disegni sullo schermo), animazioni (simulare i movimenti) e musica.

Notate che potete emettere il comando LOAD. Se digitate:

RUN APPLEVISION<RETURN>

verrà cancellato il programma presente nella memoria centrale, caricato il nuovo programma ed eseguito.

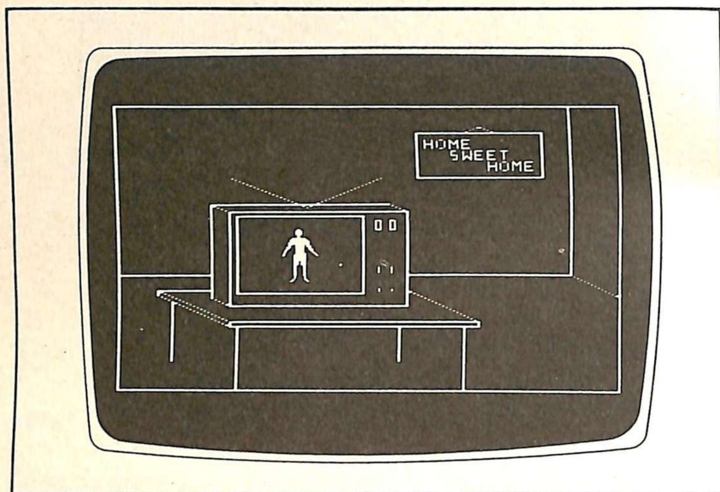


Fig. 2.6 Disegno del programma «Applevision»

Usate la seguente procedura se desiderate copiare il programma in un altro dischetto o semplicemente memorizzare il programma appena introdotto:

Introducete o richiamate con l'istruzione LOAD il programma che volete memorizzare. Controllate che il dischetto sia nel drive e digitate

SAVE nome del programma<RETURN>

dove il «nome del programma» è il nome con cui chiamate quel programma.

Questa istruzione farà copiare al DOS sul dischetto il contenuto della memoria dell'elaboratore e chiamare il file con il nome scritto dopo il comando SAVE. Per controllare che ciò sia successo digitate:

```
CATALOG
```

Troverete il nuovo file nella lista.

D'ora in poi facciamo l'ipotesi che non abbiate un drive e che usiate una registratore a cassetta e nessun altro software Apple. La versione del BASIC disponibile sarà sia Integer sia a «virgola mobile». Dipenderà dall'età del vostro sistema. Notate che se avete soltanto l'«Integer BASIC» è disponibile in cassetta anche l'Applesoft BASIC («a virgola mobile»). Infine, per concludere l'argomento «dischetto», se il vostro sistema ha un drive, potete seguire questo libro, usando la seguente procedura:

Accendete il sistema (il sistema fa «bip», il drive fa l'auotdiagnosi e sullo schermo appare la scritta «A2»). Spingete contemporaneamente «<CTRL>/<RESET>» (la scritta scompare, la luce del drive si spegne, il drive smette di ronzare e nella parte inferiore sinistra dello schermo appare il pronto del BASIC ed il cursore). Capirete quale tipo di BASIC dalla forma del pronto e del cursore. Noi ipotizziamo che abbiate il sistema di tipo recente con l'Applesoft BASIC a virgola mobile (il pronto è una parentesi quadra).

Ritorniamo al punto in cui gli utilizzatori di cassetta hanno caricato il programma chiamato «Torre». Facciamo girare questo programma digitando RUN e spingendo il tasto <RETURN>. Se avete caricato un programma di giochi, il programma stesso vi darà le istruzioni per giocare.

Queste istruzioni e tutti i messaggi tra l'elaboratore e l'utilizzatore, si chiamano «dialogo». Il dialogo completo per caricare e far girare il programma «Torre» è mostrato nelle fig. 2.7, 2.8 e 2.9.

Di solito è utile scrivere i propri programmi od i propri nastri, specialmente se incominciano ad essere lunghi ed elaborati.

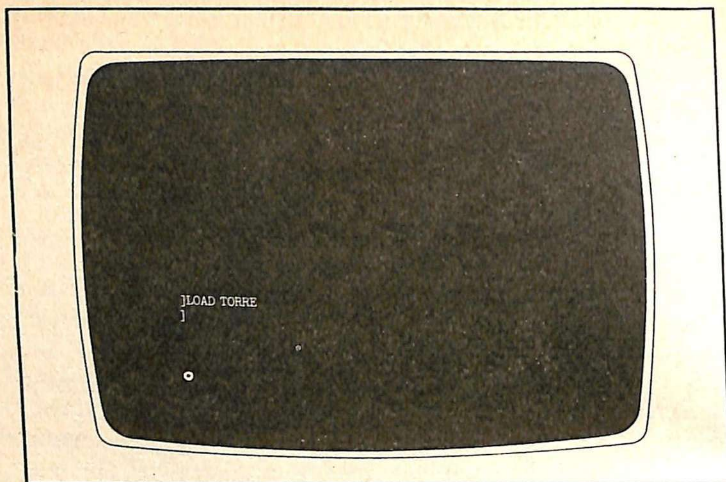


Fig. 2.7 Caricamento programma «TORRE»

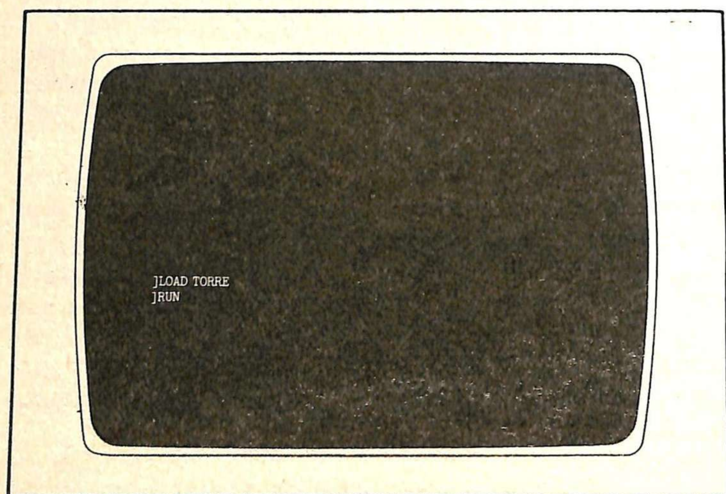


Fig. 2.8 Per far girare il programma «TORRE»

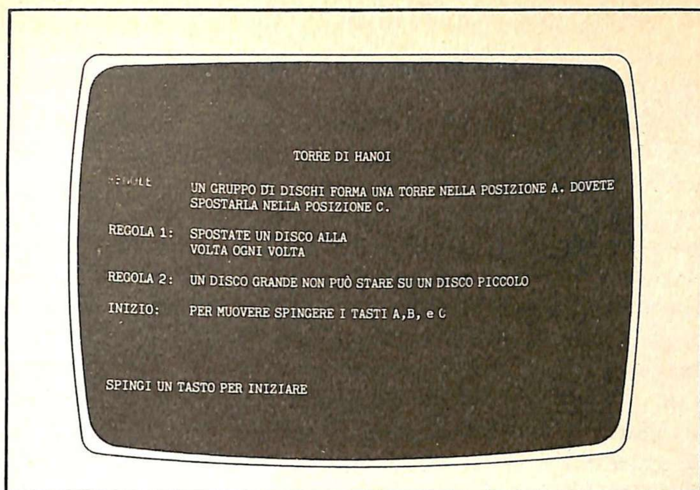


Fig. 2.9 Istruzioni per il programma «TORRE»

Quando avete terminato il programma, lo potete memorizzare con gli altri in una sola cassetta. Per indicare l'inizio di un nuovo programma, registrate a voce il messaggio: «qui comincia il programma giochi n. 2».

Editor

L'«Editor» è il nome dato alla correzione o sostituzione di una parte del programma. L'Apple ha delle semplici istruzioni di Editor. Se vi accorgete subito di aver digitato un errore, potete facilmente correggerlo usando il tasto <LEFT ARROW>. Spingendolo, il cursore torna indietro e potete digitare la lettera giusta. Se l'errore è diversi caratteri indietro spingete più volte il tasto <LEFT ARROW> finché il cursore non arrivi sull'errore e digitate correttamente.

Spingendo il tasto <RIGHT ARROW> potete ritornare al punto di prima, senza cancellare i caratteri intermedi.

Tuttavia, se premete il tasto <RETURN> quando il cursore è nel mezzo di una riga, tutti i caratteri a destra del cursore verranno cancellati. Se avete già incominciato a digitare la riga con l'errore, potete normalmente ripeterla interamente. Diciamo che state digitando un programma con la seguente riga:

110 PRINT "Cosa c'è da farre?"

Una volta digitato <RETURN> alla fine della riga, potreste non saper usare il tasto <LEFT ARROW> per correggere la parola «farre». Tuttavia avete la necessità di correggere la lettera «r» in più. Potreste decidere di ridigitare l'intera riga. Per fare ciò reinserite la riga correttamente, assicurandovi di cominciarla con il giusto numero di riga. Usate con attenzione questo metodo. Ogni volta che cominciate una riga del testo con un numero, il BASIC capirà che è una linea di programma; la nuova riga sostituirà del tutto un'altra dello stesso numero. Così se digitate

100 PRINT "cosa c'è da fare?" <RETURN>

non avete cambiato la riga sbagliata al numero 110, ma avete cambiato tutta la riga 100. La riga 110 rimarrà esattamente così com'è.

Similmente se digitate un numero di riga senza alcun testo, cancellerete tutto il testo esistente su quella riga (state ordinando all'Apple di sostituire quella riga con niente ed esattamente questo verrà fatto). Così se digitate «100» e vi accorgete di voler cambiare la riga 110 e spingete immediatamente <RETURN> per poter scrivere la riga 110, non avrete fatto altro che cancellare la riga 100.

È meglio abituarsi ad usare l'Editor dell'Apple che è un piccolo programma od una routine che vi permette di scrivere testi sullo schermo (è la forma più semplice di trattamento di testi). Purtroppo l'Editor dell'Apple è rozzo e difficile. Perciò vi raccomandiamo, se volete far dá soli un gran numero di programmi, di comprare un Editor più avanzato che vi possa aiutare. Per esempio il Writer dell'Apple. Non tratteremo qui l'uso del Writer perché, per questo, c'è un manuale specifico molto buono.

Una «modalità» è una condizione od un insieme di condizioni in base alle quali si applicano certe regole. Ci sono due modalità che dovete conoscere, l'IMMEDIATA e la DIFFERITA. Se registrate un testo nell'Apple su una riga cominciando con un numero, l'elaboratore lo memorizzerà senza agire; ciò vuol dire che siete in modalità differita. L'elaboratore registra che avete digitato una riga di un programma che volete far girare dopo, insieme a molte altre righe.

Se non usate un numero all'inizio della riga, l'Apple registrerà la modalità immediata.

In questo caso l'Apple agirà sulle istruzioni ricevute non appena spingete il tasto <RETURN>. Nel seguito di questo capitolo studieremo l'uso dell'elaboratore nella modalità IMMEDIATA. Nel capitolo 3 cominceremo a studiare la modalità DIFFERITA, cioè la programmazione.

Generalmente, ciò che un elaboratore fa, nella maggior parte dei casi, è ricevere ed immagazzinare una certa informazione, elaborarla o trattarla e poi fornire i risultati in forma appropriata. Ciò si può dimostrare ordinando all'elaboratore di fare qualcosa con una serie di caratteri. Questa serie è chiamata «stringa». Nel BASIC il simbolo $ è usato per dire all'elaboratore che volete che tratti qualcosa sotto forma di stringa. Per esempio, la stringa «NICOLA E GIOVANNI» può essere immagazzinata nella memoria dell'Apple digitando ed inserendo il seguente:

```
A$ = "NICOLA E GIOVANNI"
```

Naturalmente dovete ricordarvi di premere il tasto <RETURN>, per ordinare all'elaboratore di obbedire al comando dato. L'aver premuto <RETURN> fa eseguire all'elaboratore l'istruzione; esso infatti, dà il nome A$ ad una parte della sua memoria in cui immagazzina la stringa di caratteri compresi tra le virgolette. Ciò è mostrato nella figura 2.10. L'istruzione in linguaggio BASIC è equivalente a: «Immagazzina la stringa tra virgoletre e chiamala A$». Gli spazi e le lettere sono caratteri.

Quando premete <RETURN> ed il comando viene eseguito non c'è alcun segno esteriore che il fatto sia successo, oltre al comparire del pronto all'inizio della riga successiva. Ciò vuol dire che l'Apple è pronto a ricevere il vostro successivo comando. Tuttavia c'è ora un'area della memoria dell'elaboratore che ha memorizzato la frase «NICOLA E GIOVANNI», ed è individuata dall'elaboratore come A$.

Per dimostrare che l'istruzione è stata eseguita dovete imparare a richiamare l'informazione che è stata appena memorizzata.

Potete stampare l'informazione memorizzata in un settore della memoria chiamata A$, digitando:

PRINT A$<RETURN>

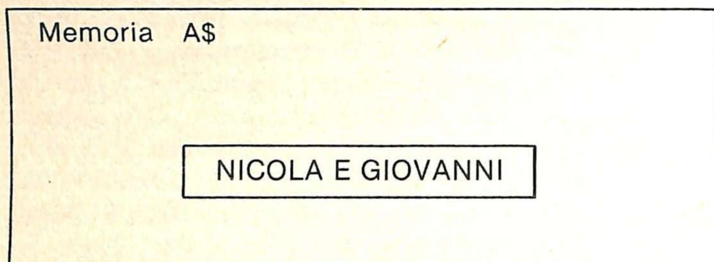


Fig. 2.10 Stringa A$ memorizzata

Come risposta l'elaboratore scriverà sullo schermo:

NICOLA E GIOVANNI

L'istruzione BASIC vuol dire «stampa quello che è memorizzato in A$».

Se dovete dare il comando all'Apple di trovare il numero di caratteri memorizzati nell'area della memoria chiamata A$, usate l'istruzione LEN, dove LEN = lenght (cioè lunghezza). Se supponiamo di digitare:

PRINT LEN (A$)

dopo aver spinto <RETURN> apparirà sullo schermo il numero 17. In questo caso i 17 caratteri sono 15 lettere e due spazi. Questa istruzione vuol dire «stampa la lunghezza della stringa memorizzata in A$» oppure «stampa il numero dei caratteri memorizzati nella stringa A$».

Il BASIC ha molti comandi per manipolare le stringhe. Il comando LEFT$ (left = sinistra) vi permette di prendere un certo numero di caratteri dalla stringa. RIGHT$ (right = destra) vi permette di prendere un certo numero di caratteri sulla destra della stringa. Per esempio digitando:

```
PRINT LEFT$(A$,6)<RETURN>
```

l'elaboratore stamperà i seguenti 6 caratteri:

```
NICOLA
```

Digitando:

```
PRINT RIGHT$(A$,8)<RETURN>
```

avrete gli otto caratteri GIOVANNI dalla stringa A$.

L'ultima istruzione vuol dire stampa gli otto caratteri sulla destra della stringa di caratteri memorizzata in A$.

Potete vedere che le istruzioni BASIC di programmazione sono delle abbreviazioni di parole scritte in Inglese. Poiché l'Apple non può pensare, tutte queste istruzioni devono essere espresse in maniera molto precisa. Se ci fosse anche un piccolo errore nello scrivere un'istruzione, l'elaboratore non sarebbe in grado di riconoscerla. L'Apple vi farà sapere di non aver compreso ciò che volete dire, stampando sullo schermo un messaggio di errore. Diamo alcuni esempi del loro significato:

?SYNTAX ERROR	(errore di sintassi. È l'errore più frequente. È normalmente uno sbaglio di battuta o la scarsa conoscenza del formato corretto dei comandi di BASIC. Se avete usato l'Integer BASIC il messaggio sarà: SYNTAX ERROR)

***>32767 ERR — (è un messaggio di errore dell'Integer BASIC. Significa che avete digitato un valore maggiore di +32767 o minore di —32767. L'Integer BASIC non può trattare valori più grandi).

I/O ERROR — (è un errore DOS. Significa che lo sportello del drive è aperto o che il dischetto non è stato inizializzato).

PROGRAM TOO LARGE — (è un errore DOS. Significa che la memoria disponibile dell'Apple non è sufficiente a contenere il file).

Abbiamo visto come il BASIC divide le stringhe. È anche facile costruirle. Per dimostrare ciò è necessario memorizzare almeno due stringhe. Digitate:

S$ = "AUTO"<RETURN>
T$ = "MOBILE"<RETURN>

Vediamo come sia possibile costruire una stringa partendo da queste due. Digitate:

PRINT S$ + T$<RETURN>

Questa istruzione significa «scrivi la stringa memorizzata in S$ seguita dalla stringa memorizzata in T$. Si forma:

AUTOMOBILE

Il seguente problema è un po' più complesso, ma cercate di capire da soli che cosa si dica di fare all'elaboratore:

PRINT LEFT$(S$,2) + RIGHT$(T$,2)<RETURN>

Il risultato è AULE.

Nello stesso modo è possibile far preparare all'Apple altre parole partendo dalle stringhe S$ e T$. Provate a far scrivere all'elaboratore le seguenti parole: «LEAU», «MOBA», «AMOB», «ABILE».

L'Apple come calcolatrice

L'Apple può essere usato come calcolatrice. Può sembrare una calcolatrice piuttosto cara, ma questo è solo uno, e forse il minore, dei modi in cui può essere usato.

I tasti dei numeri sono sulla fila più alta della tastiera. Ci sono anche cinque tasti aritmetici. Quattro sono:

+ aggiungi
— sottrai o togli
* moltiplica per
/ dividi per

la quinta funzione è la funzione aritmetica esponenziale. Il simbolo usato è « ». Questa funzione è usata per elevare un numero alla potenza di un secondo numero. Così fare il quadrato di 6 è lo stesso che farne la potenza di 2, cioè 6*6=36. La scrittura è:

6∧2

Se volete calcolare il cubo e cioè la potenza di 3 (6*6*6) usate

6∧3

Per ora tratteremo le prime quattro funzioni. Il simbolo «*» è usato per la moltiplicazione per non confonderlo con la lettera «x». Il simbolo «/» è usato perché sulla tastiera non c'è il simbolo di divisione e per poter scrivere le frazioni su una sola riga.

Le istruzioni per fare calcoli aritmetici sono del tipo:

PRINT 2+3+4<RETURN>

La risposta 9 è visualizzata immediatamente nella riga successiva. Un calcolo più complicato potrebbe essere:

PRINT (2*3+4)/5<RETURN>

La risposta è 2.

È molto importante il modo in cui si scrivono le istruzioni di calcolo perché l'elaboratore segue sempre una determinata succesione per fare i calcoli. Questa successione è quella insegnata nelle scuole e non dovrebbe far nascere problemi.

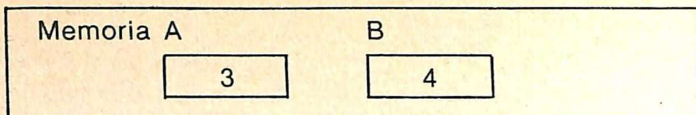


Fig. 2.11 Numeri in memoria

La somma è effettuata da sinistra a destra; moltiplicazioni e divisioni sono effettuate prima. Così:

10—2*3

Prima calcolerà 2*3 e sottrarrà il risultato da 10. La risposta è 4 e non 24. Potete cambiare questa successione usando le parentesi. Le parentesi sono sempre calcolate prima. Così se digitate

PRINT (10—2)*3<RETURN>

la risposta sarà 24. I numeri possono essere memorizzati usando le righe:

A = 3<RETURN>
B = 4<RETURN>

Per i numeri non si usa il simbolo $. Ciò che avete fatto, naturalmente, è stato di memorizzare il valore 3 nella locazione chiamata A ed il valore 4 nella locazione chiamata B. Ciò è rappresentato in fig. 2.11. Ora potete fare calcoli sui contenuti di queste locazioni senza conoscerne il valore. Ad esempio.:

PRINT A<RETURN>

dà:

3

Così:

PRINT A*B + 8<RETURN>

dà:

20

Sommario

Un buon sistema per far pratica della tastiera dell'Apple è far girare un programma di giochi che richieda risposte digitate sulla tastiera.

Un programma può essere caricato da una cassetta nella memoria dell'elaboratore, digitando il comando LOAD. Una volta caricato il programma, lo si fa girare. Quando cominciate ad usare l'Apple, è un'ottima idea far pratica con programmi già registrati su cassette (o su dischetti se avete un drive).

Le istruzioni possono essere date direttamente nel linguaggio BASIC. Con poche istruzioni è possibile far svolgere all'elaboratore attività diverse come memorizzare e manipolare parole, memorizzare numeri e fare calcoli.

Autotest

1 . Qual'è l'istruzione che fa caricare un programma nell'Apple da:
   a) una cassetta?
   b) un dischetto?
2. Come correggere una riga di programma? Cambiate la riga:

   100 L'ASCESA DELL'UOMO

   nella

   100 L'ASCESSO DELL'UOMO

   considerate di non aver ancora battuto <RETURN> alla fine della riga.
3. Memorizzate le parole «SANA» e «ROTTA». Usando soltanto queste parole memorizzate, scrivete le istruzioni che faranno stampare all'Apple:

   ROSA
   SARO
   NATTA
   TANA
4. Memorizzate i numeri 4 e 5. Usando solo questi, inserite le istruzioni necessarie per produrre i numeri 16, 24 e 36.

Capitolo 3

Introduzione alla programmazione

Scrivere e far girare un semplice programma

Verso la fine del secondo capitolo abbiamo visto l'uso dell'Apple nella modalità immediata; cioè abbiamo inserito le singole righe di istruzione da far eseguire immediatamente. In questo capitolo esamineremo la modalità DIFFERITA, che è un'altra maniera per dire «programmazione».

In tale modalità tutte le righe di istruzione dovranno cominciare con un numero. L'elaboratore memorizzerà questi comandi finché non darete il comando RUN. A questo punto eseguirà tutte le righe di istruzioni memorizzate secondo l'ordine dato dal numero di riga.

Un programma è quindi una serie di comandi che l'Apple deve eseguire. Il BASIC è il linguaggio in cui debbono essere scritti questi comandi a cui l'Apple risponderà; nel precedente capitolo abbiamo già dato alcuni esempi di comandi BASIC. Il BASIC è un semplice linguaggio di programmazione studiato nel Dartmounth College (USA) e pronto all'inizio degli anni '60. Fu preparato perché fosse facile da imparare e da disegnare e, in realtà, la sua enorme popolarità come linguaggio per microelaboratori deriva dal fatto che è facile da imparare.

L'Apple per prima si occupò di memorizzare un programma in modo da farlo eseguire quando si volesse. Quando un programma è memorizzato nella memoria dell'Apple lo si può far girare tutte le volte che si voglia o lo si può modificare prima di farlo girare di nuovo. Quando una riga BASIC è preceduta da un numero, l'Apple tratta questa riga come istruzione di un programma BASIC e memorizza sia il numero sia il comando. Il comando non è eseguito subito ma è memorizzato per essere eseguito dopo. Un programma Ap-

ple è formato da una serie di comandi numerati. I numeri danno l'ordine in cui eseguire i comandi quando il programma gira. Il comando od i comandi nella riga con il numero più basso sono eseguiti prima e così via in ordine crescente. Le istruzioni che formano un programma possono essere inserite in qualsiasi ordine giacché l'Apple usa il numero di riga per ordinare le istruzioni.

Riassumendo: una riga di programma è formata da un numero seguito da uno o più comandi; l'Apple memorizza queste istruzioni e le ordina utilizzando il numero di riga; un programma è un insieme di istruzioni; quando un programma è eseguito ogni istruzione è presa in sequenza per essere eseguita la parte del comando.

Ora scriveremo un breve programma per memorizzare le tre parole «IL», «CANE» e «GUARDA», per prendere le parole in memoria e scrivere le frasi «IL CANE GUARDA», «GUARDA IL CANE», «IL GUARDA CANE». Prima di iniziare è meglio digitare:

```
NEW<RETURN>
```

perché questo comando cancella ogni programma precedentemente memorizzato.

Digitate il seguente programma esattamente come è scritto, ogni riga, sarà memorizzata spingendo alla fine il tasto <RETURN> (in tutti i programmi di questo libro il simbolo SPC indica che dovete digitare uno spazio bianco):

```
10 A$ = "IL (SPC)"<RETURN>
20 B$ = "CANE (SPC)"<RETURN>
30 C$ = "GUARDA (SPC)"<RETURN>
40 HOME<RETURN>
50 PRINT A$ + B$ + C$<RETURN>
60 PRINT C$ + A$ + B$<RETURN>
70 PRINT A$ + C$ + B$<RETURN>
```

In questo programma le tre parole sono memorizzate nelle righe 10, 20 e 30. È inserito uno spazio alla fine di ogni parola per separarle quando si scriverà la frase. La riga 40 pulisce lo schermo e pone il cursore in alto a sinistra quando girerà il programma. HOME è il comando dell'Apple che cancella

completamente lo schermo. È una buona regola generale avere l'abitudine di usare questo comando all'inizio di tutti i vostri programmi.

Le righe 50, 60 e 70 fanno stampare le frasi richieste. La fig. 3.1 mostra il risultato dell'esecuzione delle istruzioni del programma. Il risultato di un programma è la somma dei risultati dell'esecuzione di tutte le istruzioni.

Per controllare che il programma sia stato memorizzato bene nell'Apple digitate

LIST<RETURN>

l'elaboratore mostrerà un «listato» del programma in memoria. Controllate il listato sullo schermo dell'Apple confrontandolo con quello scritto precedentemente per vedere se sia esatto, se così fosse eseguite il programma digitando:

RUN<RETURN>

Vedrete il risultato del programma che deve essere:

```
IL CANE GUARDA
GUARDA IL CANE
IL GUARDA CANE
]
```

Ricordate che potete far eseguire o listare il programma a vostro piacimento poiché è memorizzato nell'Apple.

Se volete listare solo una parte od addirittura solo una riga del programma digitate «LIST da x, a x» oppure LIST ed il numero della riga. Se il programma è stato inserito con errori potrà essere corretto listandolo sullo schermo per individuare l'errore e digitando la nuova riga. Ad esempio, se la riga 20 fosse stata sbagliata potrebbe essere listata digitando:

LIST 20<RETURN>

PROGRAMMA	CONTENUTO DELLA MEMORIA DOPO L'ESECUZIONE DELLE ISTRUZIONI	ELABORAZIONE INTERMEDIA	STAMPA SUL VIDEO
10 A$ = "IL"	A$ IL		
20 B$ = "CANE"	B$ CANE		
30 C$ = "GUARDA"	C$ GUARDA		
40 HOME			Pulisci lo schermo
50 PRINT A$ + B$ + C$		A$ + B$ + C$ IL CANE GUARDA	IL CANE GUARDA
60 PRINT C$ + A$ + B$		C$ + A$ + B$ GUARDA IL CANE	GUARDA IL CANE
70 PRINT A$ + B$ + C$		A$ + C$ + B$ IL GUARDA CANE	IL GUARDA CANE

Fig. 3.1 Il risultato del programma che gira

vi potrebbe apparire:

20 B$ = "CNE"

Siccome è una riga corta, per correggerla rapidamente è più semplice ridigitare la linea intera. Se avete l'intenzione di scrivere molti programmi o pensate di usare l'elaboratore per produrre documenti vi conviene acquistare un editor di testi.

Per cancellare completamente una riga senza editor è sufficiente digitare il numero di riga da cancellare seguito da <RETURN>.

Provate digitando:

60<RETURN>

il listato del programma vi mostrerà cosa ha provocato. Quando avete finito di provare il programma digitate:

NEW<RETURN>

per cancellarlo. Dopo aver fatto questo il comando LIST non avrà risposta. Provate.

Alcune istruzioni BASIC

In questo paragrafo incontreremo altre istruzioni BASIC. Sono inserite in un breve programma che ne mostra l'utilità. Non diremo ogni volta della necessità di spingere il tasto <RETURN> dopo un comando ed alla fine di un'istruzione perché l'Apple II agisca. Così, per l'ultima volta, ricordiamo che se avete digitato qualcosa e non appare niente di quello che vi aspettate, vuol dire che non avete spinto il tasto <RETURN>.

Input

Immaginiamo di voler modificare il programma scritto nel primo paragrafo di questo capitolo, in modo che, quando gira, accetti tre parole qualsiasi e stampi la prima, seguita dalla seconda e dalla terza; quindi la terza, seguita dalla prima e della seconda ed infine la seconda, seguita dalla prima e dalla terza. La nuova istruzione necessaria perché il programma accetti queste parole è INPUT. Quando è eseguita, una istruzione INPUT fa stampare sullo schermo un punto interrogativo per indicare che è richiesta una risposta, ferma l'esecuzione del programma ed ordina all'Apple di aspettate finché l'utilizzatore non digiti una risposta che dovrà accettare. Così l'istruzione:

```
10 INPUT A$
```

causa la comparsa sullo schermo di un punto interrogativo.

Se voi digitate:

IL<RETURN>

la parola «IL» è presa e memorizzata in A$. Così, in questo caso, il risultato è lo stesso di quello che si ottiene con l'istruzione

10 A$ = "IL"

la differenza è che l'ultima assegna alla variabile A$ la parola IL mentre la precedente assegna qualsiasi cosa voi digitiate dopo il punto interrogativo.

L'esempio si basa sul precedente programma con la sostituzione delle righe che memorizzano le tre parole con tre istruzioni INPUT (una per ogni parola). Dopo aver registrato le parole in questo modo, le righe da 50 a 70 stamperanno le frasi come prima. Non è però possibile introdurre uno spazio dopo le parole memorizzate con il comando INPUT. Per quasto motivo lo spazio per separare le parole in una frase deve essere previsto nell'istruzione PRINT. Abbiamo così ottenuto il seguente programma per memorizzare tre parole e stampare con esse tre frasi.

```
10 HOME
20 INPUT A$
30 INPUT B$
40 INPUT C$
50 PRINT A$ + "(SPC)" + B$ + "(SPC)" + C$
60 PRINT C$ + "(SPC)" + A$ + "(SPC)" + B$
70 PRINT B$ + "(SPC)" + A$ + "(SPC)" + C$
```

Quando sarà eseguito il programma, si avrà questo dialogo:

```
? IL
? LAVA
? CANE
IL LAVA CANE
CANE IL LAVA
LAVA IL CANE
]
```

Decisioni

L'Apple può essere programmato per prendere decisioni. Questa possibilità può essere usata per scrivere programmi molto interessanti e potenti. Il comando che permette di prendere decisioni usa i termini del BASIC IF e THEN. Ha la forma:

```
IF condizione THEN comando
```

Nella parte «condizione» di questa istruzione vi possono essere variabili e/o valori e naturalmente si vede se sono eguali o diversi. La parte «comando» sarà un comando BASIC, ad esempio un'istruzione di assegnazione od un comando PRINT. Quando viene eseguito, un comando IF/THEN prima controlla la parte «condizione». Solanto se questa prova sarà positiva sarà eseguita la parte «comando». Se la parte condizione non fosse vera, la parte «comando» sarà ignorata. Un esempio è:

```
IF N$="PAROLACHIAVE" THEN PRINT "ACCETTATO"
```

Quando viene eseguita, l'Apple controlla che l'ultima parola assegnata ad N$ sia PAROLACHIAVE; se questo è vero sarà stampato ACCETTATO, se falso non accade nulla. Un secondo esempio è:

```
IF N$<>"PAROLACHIAVE" THEN PRINT "RESPINTO"
```

In questo esempio i simboli <> significano «non eguale». Così quando si esegue il programma, sarà stampato RESPINTO soltanto se l'ultima parola assegnata ad N$ non è PAROLACHIAVE.

Adesso vediamo un breve programma per indicare un'addizione, scriverla, accettare la risposta e decidere se sia giusta o meno prima di stampare il messaggio più appropriato.

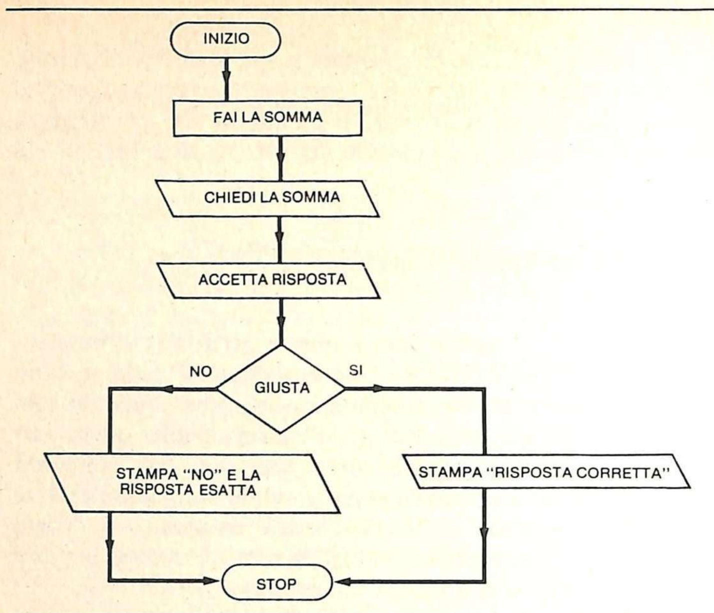


Fig. 3.2 Diagramma di flusso

Questo è anche mostrato in figura 3.2 che è un esempio di «diagramma di flusso». Il programma inizia memorizzando i due numeri A e B e la riga 30 li usa per stampare le domande «qual'è la loro somma?».

La domanda da stampare è del tipo QUANTO VALE 2+3? Se si usa un punto e virgola per separare le parti di un'istruzione PRINT, questa lascerà uno spazio diverso da quello che lascerebbe usando la virgola. Avendo posto la domanda il programma accetterà una risposta alla riga 40 e la memorizzerà in C. Alla riga 50 si confronterà la risposta con il risultato dell'addizione e se coinciderà sarà stampato un messagio di incoraggiamento. L'ultima riga è usata quando è stata data una risposta sbagliata e fa stampare la risposta giusta. Il programma è:

```
10 A=2
20 B=3
```

```
30 PRINT "QUANT'È (SPC)"; A;" + ";B;"?"
40 INPUT C
50 IF C=A+B THEN PRINT "BENE. È GIUSTO"
60 IF C<>A+B THEN PRINT "NO. LA RISPOSTA È
";A+B
```

Questo programma può essere modificato per dare più di una possibilità di risposta giusta, al contrario del programma della Fig. 3.2, usando il comando GOTO.

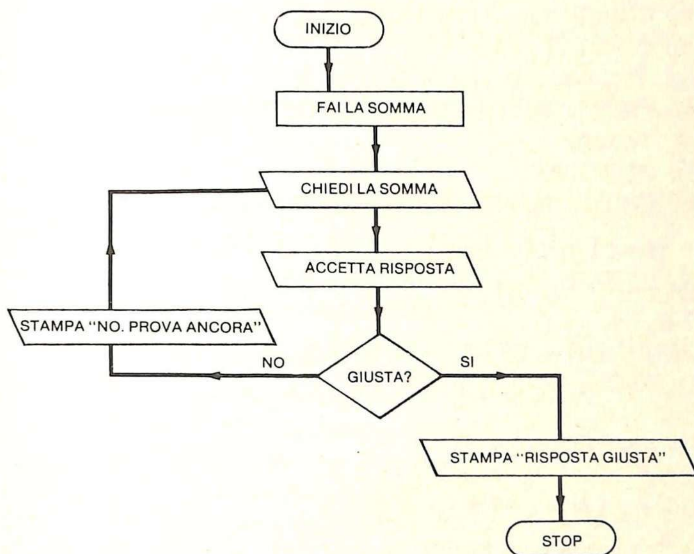


Fig. 3.3 Diagramma di flusso modificato

Il comando:

GOTO 30

ordina al programma di andare alla riga 30 ed eseguirla. Un programma che ripresenta la domanda finché non sia

stata data la risposta giusta si ottiene modificando la riga 50 in modo che l'ultima riga del programma (riga 80) sia eseguita solo dopo la risposta giusta. L'ultima riga fa stampare una frase di incoraggiamento. Se non è stata data una risposta giusta è eseguita la riga 60 che stampa che la risposta è sbagliata, prima di eseguire la riga 70 che fa saltare il programma alla riga 30, così la domanda è formulata di nuovo e si dà la possibilità di un'altra risposta. Il listato di questo programma più complicato è:

```
10 A=2
20 B=3
30 PRINT "QUANT'È (SPC)"; A;" + ";B;"?"
40 INPUT C
50 IF C=A+B THEN GOTO 80
60 PRINT "SPIACENTE. RISPOSTA SBAGLIATA.
RIPROVA"
70 GOTO 30
80 PRINT "BENE. RISPOSTA GIUSTA".
```

Si avrà questo dialogo:

```
QUANT'È 2+3?
? 6
SPIACENTE. RISPOSTA SBAGLIATA. RIPROVA.
QUANT'È 2+3?
? 4
QUANT'È 2+3?
? 5
BENE. LA RISPOSTA È GIUSTA.
```

Ora cambiamo la riga 30 con:

```
30 PRINT "QUANT'È (SPC)";A;" + ";B;
```

Questa volta non abbiamo messo il punto interrogativo alla fine. Inoltre il punto e virgola alla fine della riga «spinge in alto» il punto interrogativo dell'istruzione INPUT. Abbiamo così eliminato il doppio punto interrogativo per aver una successione di domanda e risposta più accettabile. Si avrà questo dialogo:

```
QUANT'È 2+3? 6
SPIACENTE. RISPOSTA SBAGLIATA. RIPROVA
```

```
QUANT'È 2+3? 4
SPIACENTE. RISPOSTA SBAGLIATA. RIPROVA
QUANT'È 2+3? 5
BENE. RISPOSTA GIUSTA
```

Ripetizione

Il precedente programma ha mostrato che l'Apple può essere programmato per fare la stessa cosa molte volte usando il comando GOTO. Il quesito aritmetico è posto più volte finché non viene data la risposta giusta. Il BASIC ha un metodo più rapido per ottenere lo stesso effetto: il comando FOR...NEXT. Per vederne l'uso, digitate il seguente programma:

```
10 FOR I=1 TO 16
20 PRINT "LAURA"
30 NEXT I
```

Questo programma farà stampare 16 volte «LAURA» perché tutte le istruzioni comprese tra FOR e NEXT sono ripetute tante volte così come indicato dal comando FOR. La riga 10 del nostro esempio incrementa un contatore ogni volta che è eseguito il comando PRINT «LAURA». Il programma cammina in cerchio e questo è detto «LOOP».

Ogni volta che è eseguita, la riga 30 ordina all'elaboratore di tornare alla riga 10 finché il contatore non raggiunga il numero scritto nell'istruzione TO, cioè 16. A questo punto il programma esce dal loop ed esegue, se esiste, il comando successivo. Il prossimo programma mostra che potete inserire tutte le istruzioni che volete tra le istruzioni FOR e NEXT:

```
10 FOR I=1 TO 9
20 PRINT "RIPETIZIONE NUMERO (SPC)(SPC)";K
30 PRINT "FRANCESI"
40 PRINT
50 NEXT I
```

La ripetizione avviene 9 volte e avremo l'output:

```
RIPETIZIONE NUMERO 1
FRANCESI
```

RIPETIZIONE NUMERO 2
FRANCESI
....

e continua fino alla ripetizione numero 9.

Proviamo un programma che memorizza una parola e la divide lettera per lettera. Il programma memorizza la parola, ne calcola la lunghezza, la divide ripetutamente e stampa la prima lettera, la seconda e così via sino all'ultima. Abbiamo già visto come si dividono le lettere a sinistra ed a destra di una parola usando LEFT$ e RIGHT$. Il BASIC può anche esaminare la parte centrale di una parola con il comando MID$. Si usa così:

```
MID$(N$,2,4)
```

prende quella parte della parola assegnata a N$ che inizia dalla seconda lettera e sia lunga quattro lettere. Ad esempio:

```
J$ = "GIOVANNA"
PRINT MID$(J$,5,4)
```

darà "ANNA", e:

```
F$ = "FRANCESCO"
PRINT MID$(F$,3,4)
```

darà "ANCE"

Conoscete così i comandi necessari per scrivere il programma. Incomincerà con un'educata richiesta di scrivere un nome, seguirà un comando di INPUT che memorizza la parola in W$. Alla riga 30 si calcola il numero di lettere della parola e lo si memorizza il L. La riga 50 ha MID$(W$,I,1) con I che è la variabile del loop del comando FOR...NEXT che è lungo quanto la lunghezza della parola. In altre parole I è la prima lettera della parola, la seconda e così via. MID$(W$,3,1) troverà la stringa di lettere della parola memorizzata in W$ che inizia dalla terza lettera della parola ed è lunga una sola lettera, cioè, in altre parole, troverà la terza lettera della parola. Le righe da 40 a 60 servono a trovare le lettere della parola ed a stampare, nel primo loop, «LETTE-

RA NUMERO 1» seguita dalla prima lettera e così via. L'intero programma che divide la lettura di una parola è:

```
 5 HOME
10 PRINT "PER FAVORE, SCRIVETE UNA PAROLA"
20 INPUT W$
30 L = LEN(W$)
40 FOR I = 1 TO L
50 PRINT "LETTERA, NUMERO(SPC)";I;"(SPC)(SPC)";
MID$(W$,I,1)
60 NEXT I
```

Altri programmi

Scriviamo ora un programma che accetti ogni nome scritto nella forma:

```
MARIO ROSSI
```

e che produca l'output:

```
IL VOSTRO NOME È MARIO
IL VOSTRO COGNOME È ROSSI
```

Potrebbe sembrare molto facile, così:

```
INPUT N$
```

il nome può essere trovato come LEFT$(N$,5) ed il cognome come RIGHT$(N$,5). Purtroppo questo modo darà risposte assurde nel caso si digiti GUGLIELMO DUCALE (oppure ogni nome che non sia composto da nome e cognome di cinque lettere). Il trucco consiste, ovviamente, nell'individuare lo spazio che separa il nome dal cognome. Così, nell'ipotesi che il nome ed il cognome siano stati digitati bene, tutto ciò che è alla sinistra dello spazio sarà il nome, alla destra il cognome. Se il nome ed il cognome non saranno digitati in questo modo, potete aspettarvi di veder visualizzato un risultato strano, così il programma dovrà domandare di inserire nome e cognome in modo standard, controllare che ciò avvenga e rifiutarlo se non fosse correttamente inserito. Questo ragionamento vi porta al seguente programma:

```
 10 HOME
 20 PRINT "INSERITE IL VOSTRO NOMINATIVO.
DIGITATE"
 30 PRINT "IL VOSTRO NOME, UNO SPAZIO E"
 40 PRINT "IL VOSTRO COGNOME."
 50 INPUT N$
 60 L = LEN(N$):C = O
 70 FOR I = 1 TO L
 80 IF MID$(N$,I,1) = "(SPC)" THEN C = C + 1
 90 NEXT I
100 IF C<> 1 THEN PRINT "PER FAVORE DIGITATE IL
NOMINATIVO COME RICHIESTO"
110 IF C<> 1 THEN GOTO 20
120 FOR J = 1 TO L
130 IF MID$(N$,J,1) = "(SPC)" THEN B = J
140 NEXT J
150 PRINT "IL VOSTRO NOME È (SPC)";LEFT$(N$,B—1)
160 PRINT "IL VOSTRO COGNOME È (SPC)";RIGHT$(N$,L—B)
```

Il programma all'inizio pulisce il video, dopo le righe da 20 a 40 mostra sullo schermo le istruzioni per usare il programma. La riga 50 chiede un nominativo che memorizza in N$. La riga 60 ha due comandi separati da due punti; il primo comando memorizza in L il numero di caratteri digitati, il secondo pone a zero C che servirà per contare il numero di spazi. Le righe da 70 a 90 scandiscono la lettura del nominativo contando il numero di spazi. Alla fine del loop il numero di spazi è memorizzato in C. Se il numero di spazi non è esattamente 1, le righe 100 e 110 indicano che il nominativo non è stato digitato correttamente e fanno tornare alla riga 20 per permetterevi di digitare di nuovo il nominativo. Le righe da 120 a 140 individuano la posizione dello spazio, la memorizzano in B, così la riga 150 visualizzerà come nome tutti i caratteri a sinistra dello spazio, mentre la riga 160 visualizzerà come cognome tutti i caratteri a destra dello spazio.

Il prossimo programma produce una visualizzazione molto divertente di un oggetto a forma di verme che si muove

sullo schermo avanti ed indietro. Benché ci siano altri modi per produrre lo stesso effetto, questo metodo introduce un nuovo modo di scrivere in determinate posizioni dello schermo. HTAB (tabulatore orizzontale) individua la colonna, VTAB (tabulatore verticale) individua la riga. Il massimo valore HTAB per riga è 40 (HTAB 41 sposta il cursore sulla prima colonna della riga successiva). Il massimo valore di VTAB è 24; ogni valore superiore produrrà il messaggio «?ILLEGAL QUANTITY ERROR». Il valore massimo assoluto di HTAB è 255. Nella riga 20 c'è un altro comando nuovo. SPEED = n cambia la velocità con cui sono visualizzati i caratteri sullo schermo secondo il calore di n. Di norma il valore massimo è 255. Ponendo SPEED = 200 abbasserete la velocità con cui il verme attraversa lo schermo. Provate questo comando assegnando diversi valori al comando SPEED.

Abbiamo anche introdotto una estensione del comando FOR...NEXT. Alla riga 70 abbiamo FOR...NEXT...STEP. Se non scrive STEP, l'elaboratore automaticamente lo assume eguale ad 1. In altre parole il contatore di FOR...NEXT aumenterà di uno ad ogni giro. Nella riga 70 STEP è —1 ed il contatore, perciò, diminuirà di uno ad ogni giro.

La forma del verme creata dal programma è definita alla riga 50 ed alla riga 100 ed è preceduta e seguita da uno spazio. Il programma si divide in due parti. Le righe da 30 a 60 fanno spostare il verme sullo schermo da sinistra a destra. le righe da 70 a 110 lo spostano nell'altro senso. Nel comando FOR della riga 30 non vi è alcun STEP così J è aumentato di 1 da 1 a 24. Nella riga 70 è incluso lo STEP, così il valore di J, che nel primo loop aumentava di 1, nel secondo loop diminuisce di 1 finché il verme raggiunge il margine sinistro. Il GOTO della riga 30 fa tornare il verme al primo loop. Se si tolgono gli spazi intorno al verme della riga 50, il verme lascerà una traccia quando si muoverà da destra a sinistra che cancellerà quando si muoverà nel senso opposto. Il programma è:

```
10 HOME
20 SPEED = 200
```

```
30 FOR J = 1 TO 24
35 VTAB 1
40 HTAB J
50 PRINT "(SPC)***(SPC)"
60 NEXT J
70 FOR J = 24 TO 1 STEP —1
80 VTAB 1
90 HTAB J
100 PRINT "(SPC)***(SPC)"
110 NEXT J
120 GOTO 30
```

Poiché l'ultima riga del programma fa eseguire di nuovo la riga 30 facendo iniziare una nuova passeggiata del verme avanti ed indietro sullo schermo, il programma, quando viene eseguito, gira all'infinito. per fermarlo è necessario premere i tasti <CTRL> e C.

Speriamo che vogliate provare tutti i programmi di questo libro e fare dei cambiamenti per vedere cosa provocano. Nell'ultimo programma ad esempio, provate un nuovo comando: RND(1). Questo comando fa produrre all'Apple un numero decimale casuale compreso tra 0 ed 1. Per avere un numero casuale compreso tra 0 e 10 è sufficiente moltiplicarlo per 10 usando il comando INT che toglie la parte decimale. Aggiungiamo 1 per rendere il numero casuale variabile tra 1 e 10. Digitate il seguente programma formato da una sola riga e vedete cosa succede quando lo fate girare:

```
100 PRINT INT(10*RND(1)) + 1:GOTO 100
```

Usiamo <CTRL> per fermarlo. Ora cambiamo la riga per leggere:

```
100 PRINT INT((240*RND(1))/10) + 1:GOTO 100
```

Questa volta si avrà una lista di numeri casuali compresi tra 1 e 24; e cioè una generazione casuale di valori che possono essere usati per VTAB.

Cambiamo l'ultimo programma con il seguente e cercate di capire con precisione cosa faccia ora il programma:

```
10 HOME
```

```
20 SPEED = 200
25 V = INT((240*RND(1)/10) + 1
30 FOR J = 1 TO 24
35 VTAB 1
40 HTAB J
50 PRINT "(SPC) ***(SPC)"
60 NEXT J
70 FOR J = 24 TO 1 STEP —1
80 VTAB 1
90 HTAB J
100 PRINT "(SPC) ***(SPC)"
110 NEXT J
120 GOTO 30
```

Fate girare il programma ed osservate ciò che succede. Cancellate la riga 115 e controllate cosa avvenga e notate le differenze.

Prepariamo ora un programma per tradurre in inglese parole italiane. Per fare ciò sarà necessario memorizzare parole italiane e le parole inglesi corrispondenti, così che si possano collegare l'una all'altra.

Il BASIC ha il «vettore» che è necessario per fare ciò. Il comando BASIC:

```
DIM A$(20)
```

informa l'Apple che venti variabili formano un vettore con i nomi che vanno da A$(1) ad A$(20). Una volta definite, queste variabili-vettore possono essere usate come nomi variabili. Ad esempio possiamo dare l'istruzione:

```
A$(6) = "MAN"
```

Il comando DIM ha già riservato lo spazio di memoria per tutte le variabili del vettore. I vettori, come mostra il prossimo programma, possono essere inseriti con grandi vantaggi nel loop FOR...NEXT. In questo programma useremo due vettori (vedi Figura 3.4); l'uno (I$) contiene le parole italiane, e l'altro (E$) contiene le equivalenti parole inglesi nello stesso ordine.

Il programma traduce cercando la parola italiana da tradurre tra quelle memorizzate in I$. Se la parola è trovata, è

individuata la parola inglese che sta nella stessa posizione del vettore E$. Nel programma, la riga 10 riserva lo spazio per due vettori che memorizzeranno le parole italiane ed inglesi. Le parole sono memorizzate con le righe da 20 a 50. Quando sono eseguite le righe fino alla 50, lo stato della memoria è mostrato in fig. 3.4. La riga 60 chiede di digitare la parola italiana e la riga 70 la prende e la memorizza. La riga 80 pone T uguale a zero: sarà uguale a zero solo se non fosse trovata la parola italiana che non potrà essere tradotta. Le righe tra 90 e 120 controllano tutto il vocabolario italiano memorizzato per trovare la parola; se ciò avviene, la riga 100 fa stampare la corrispondente parola inglese e, nella riga 110, T è cambiato in uno. Dopo questo loop, la riga 130 controlla T: se T fosse zero, la parola può essere tradotta ed è stampato un adeguato messaggio. La riga 140 fa eseguire la riga 60; così si può inserire una nuova parola italiana e la successiva parte di programma sarà ripetuta. Il programma è questo:

```
5 HOME
10 DIM E$(4), I$(4)
20 E$(1) = "MAN": I$(1) = "UOMO"
30 E$(2) = "WOMAN": I$(2) = "DONNA"
40 E$(3) = "BOY": I$(3) = "RAGAZZO"
50 E$(4) = "GIRL": I$(4) = "RAGAZZA"
60 PRINT "SCRIVI LA PAROLA ITALIANA"
70 INPUT W$
80 T = ∅
90 FOR I = 1 TO 4
100 IF W$ = F$(I) THEN PRINT E$(I)
110 IF W$ = F$(I) THEN T = 1
120 NEXT I
130 IF T = ∅ THEN PRINT W$;"(SPC) LA PAROLA
NON È NEL MIO VOCABOLARIO"
140 GOTO 60
```

Chiaramente il programma ha un vocabolario molto limitato, ma lo si può ampliare facilmente (aggiungendo diverse righe del tipo di quelle tra 20 e 50 e con altre modifiche di

poco conto). Non è neanche difficile modificare il programma perché traduca dall'inglese in italiano.

	E$(1)	E$(2)	E$(3)	E$(4)
E$	MAN	WOMAN	BOY	GIRL
	I$(1)	I$(2)	I$(3)	I$(4)
I$	UOMO	DONNA	RAGAZZO	RAGAZZA

Fig. 3.4 I due vettori paralleli per il programma traduzione

Come abbiamo già detto, tutti questi programmi servono per fare delle prove di programmazione con il BASIC; possono essere modificati, estesi ed ampliati in molti modi.

Memorizzare i programmi

Quando si spegne l'Apple, si perdono i programmi contenuti nella memoria. Per evitare di digitare il programma tutte le volte che lo si voglia usare, è necessario caricarlo in una memoria permanente. Un programma in memoria nell'Apple può essere memorizzato sia su un nastro sia su un dischetto (se disponibile il drive) per essere poi caricato di nuovo in memoria.

Per memorizzare il programma nella cassetta, bisogna prima controllare che il registratore sia connesso correttamente all'Apple e che vi sia la cassetta (preferibilmente non incisa). Riavvolgete completamente il nastro e fatelo girare un poco in avanti per poter registrare il titolo in testa al programma. Quando il nastro è al punto giusto, digitate:

```
SAVE
```

Spingete i tasti PLAY e RECORD del registratore contemporaneamente. Spingete RETURN ed il programma sarà copiato e registrato sul nastro. Il programma è ancora contenuto nella memoria dell'Apple e non è stato modificato dal comando SAVE.

Se avete un drive, potete memorizzare un programma sul dischetto con la procedura seguente. Per prima cosa controllate il dischetto da usare ed assicuratevi che non sia «pro-

tetto da scrittura». Prendete il dischetto con delicatezza in modo che la targhetta sia nell'angolo in alto a destra del dischetto. Sul lato sinistro, a pochi centimetri dall'angolo, vi è una piccola tacca. Se questa tacca è ricoperta da una targhetta color bianco ed argento, allora non vi potete copiare nulla. È questo un metodo per proteggere le informazioni già memorizzate. Se il dischetto è stato «protetto da scrittura», può diventare «a scrittura permessa» togliendo la targhetta, permettendovi di memorizzare il nuovo programma.

Accendete il drive, inserite il dischetto e caricate DOS. Controllate che il DOS sia stato caricato correttamente, digitando:

CATALOG <RETURN>

Vedrete una lista di tutti i file presenti nel dischetto. Ora memorizzate il nuovo programma oppure caricatene e modificatene uno esistente. Digitate:

SAVE

seguito dal nome che avete dato al programma.

Se fosse il vostro ultimo esempio, lo potremmo chiamare «LESSICO». Digitate:

SAVE LESSICO <RETURN>

La luce del drive si accenderà ed il drive emetterà un ronzio finché l'Apple caricherà il contenuto della memoria sul dischetto sotto il nome «LESSICO». Quando la luce si spegne, controllate, digitando:

CATALOG <RETURN>

Potrete vedere che «LESSICO» è stato aggiunto alla lista. Potete usare in sequenza: comandi LOAD e SAVE per copiare un file su un altro disco, per far copia di sicurezza dei vostri programmi.

Uso della stampante

Aggiungere una stampante all'Apple migliora moltissimo la possibilità di uso. All'inizio il principale uso della stampante è scrivere il listato di un programma ed i risultati in

forma permanente e trasportabile. Il listato di un programma non è una memoria conveniente ma può essere portato via dall'Apple e studiato con comodo. Se un programma produce molti risultati, è più comodo stamparli che copiarli dallo schermo; è anche più sicuro.

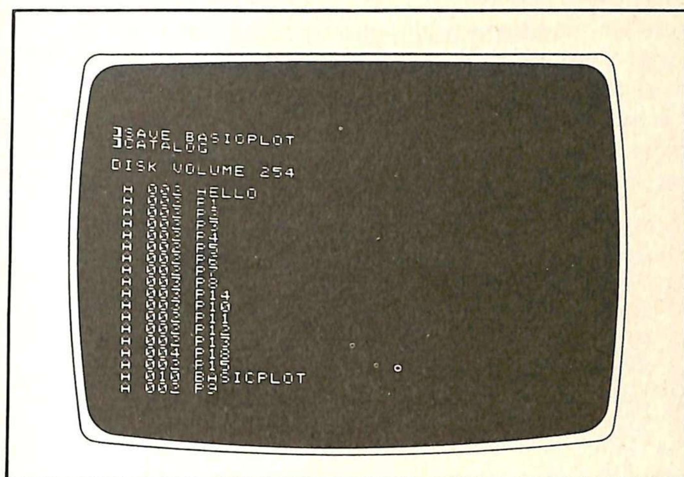


Fig. 3.5 Programma «Applevision»

Dopo gli iniziali controlli per assicurarsi che la stampante sia collegata all'Apple, sia accesa e con la carta inserita ed «in linea» con l'elaboratore, il programma memorizzato nell'Apple può esser listato sulla stampante, piuttosto che sullo schermo, col semplice comando:

PR#n

dove «n» è la porta dell'interfaccia nell'Apple che contiene il controllo della stampante. Normalmente è la porta 1. Se la vostra interfaccia è su una porta differente, dovrete inserire il numero relativo. È inutile dire che se non avete una cartolina di controllo della stampante, non sarete in grado di usare una stampante. Una volta stampato PR#1, tutto ciò che appare sullo schermo sarà anche stampato in copia.

Potete disinserire la stampante in qualsiasi momento, digitando «PR#∅». Entrambi i comandi possono essere dati nelle modalità IMMEDIATA e DIFFERITA. Se collegate la stampante da programma, solo l'output del programma sarà inviato alla stampante; in altre parole, c'è solo una attivazione temporanea della stampante.

Per mostrare come si possa usare la stampante da programma, modifichiamo il semplice programma all'inizio di questo capitolo. Si avrà lo stesso output sullo schermo, come prima, e lo si avrà, identico sulla stampante.

L'unica aggiunta che serve è aggiungere il comando per aprire il canale con la stampante: fatelo alla riga 40 all'inizio. Il nuovo programma è:

```
 5 HOME
10 A$ = "IL"
20 B$ = "CANE"
30 C$ = "GUARDA"
40 PR  1
50 PRINT A$ + B$ + C$
60 PRINT C$ + A$ + B$
70 PRINT A$ + C$ + B$
```

Badate che se cercate di usare una stampante quando non c'è nulla collegato all'Apple, il risultato probabile è un «blocco dell'elaboratore». Un blocco dell'elaboratore generalmente produce uno schermo inattivo che non accetta nessun altro comando dalla tastiera; cioè non fa nulla, e nulla farà. Vostra unica risorsa è premere <CTRL> / <RESET> ed evitare di rifare lo stesso errore. Notate che avrete lo stesso risultato se cercate di caricare da un registratore a cassette, quando siete collegati ad un dischetto.

Sommario

L'Apple può memorizzare un programma BASIC che può esser utilizzato quando si voglia, o modificato prima di farlo girare nuovamente. Il linguaggio BASIC dell'Apple è semplice come l'Inglese e, fra l'altro, offre possibilità di memorizzare e manipolare informazioni, prendere decisioni, ri-

petere un'azione ogni qualvolta sia necessario. In questo capitolo, tali possibilità sono inserite in semplici programmi che ne mostrano l'uso. Quando un programma è stato scritto, può esser conservato su cassetta o disco; abbiamo mostrato come ciò sia possibile.

Autotest

1. Qual'è il comando per avviare la procedura per conservare il programma memorizzato nell'Apple su cassetta?
2. Quali sono le parole usate in BASIC per:
   (a) ripetizione
   (b) elaborare un testo e modificare il risultato
   (c) dare istruzioni a un programma mentre gira?
3. Scrivete brevi programmi per il seguente:
   (a) stampare il vostro nome 10 volte
   (b) introdurre una parola e decidere se ha più di 7 caratteri. In tal caso indicare che è stata inserita una parola lunga; altrimenti stampare che era corta.
   (c) prendere le parole digitate sulla tastiera e stamparle senza la prima o l'ultima lettera.
4. Spiegate come illustrato in fig. 3.1 i calcoli fatti quando vengono elaborati i seguenti programmi:

(a)
```
10 A$ = "ALGORITMO"
20 L = LEN (A$)
30 FOR I = 1 TO L
40 PRINT LEFT$(A$,I)
50 NEXT I
```
(b)
```
10 A = 1:B = 1
20 PRINT A: PRINT B
30 FOR I = 1 TO 12
40 C = A + B
50 PRINT C
60 A = B: B = C
70 NEXT I
```

5. Scrivete un programma per prendere una parola, memorizzarla in A$ e memorizzare in B$ il suo inverso. Si può fare iniziando con una stringa di zero caratteri in B$, aggiungendo poi un carattere alla volta, dalla destra di A$. Il programma stamperà la parola inversa e poi stabilirà se la parola originale sia un palindromo, cioè se si legga in entrambi i sensi.

Un tipico dialogo per il programma potrebbe essere:

```
INTRODUCI UNA PAROLA, PER FAVORE
? ADA
L'INVERSO DI ADA È ADA
ADA È UN PALINDROMO
```

Capitolo 4

Grafici

Nella lingua degli elaboratori i grafici sono disegni. Molti programmi, scritti per l'Apple, che sono di notevole interesse e valore, si serviranno spesso di grafici. Ciò è in particolar modo indicato per programmi di educazione (vedi fig. 1.8) e di giochi, nei quali l'interesse e la necessità dei programmi spesso consistono nella capacità di attrazione dei grafici. Anche i programmi d'affari possono aver molto più risalto se offrono dati e risultati sia con disegni sia in forma numerica.

Naturalmente il calcolo numerico è necessario in qualsiasi programma ragionevolmente complesso, qualunque sia la sua applicazione, ma il suo risultato può esser rappresentato in uno di questi tre modi: numeri, parole, disegni. Mentre in alcune applicazioni è necessario avere accurati risultati numerici, in molte altre la presentazione di uno schermo pieno di numeri, inevitabilmente, prima o poi, diventa monotona. Presentare dati per mezzo di parole è meglio che con i soli numeri, ma è più facile leggere da un libro, che da uno schermo! Comunque, come è risaputo, un disegno vale mille parole e le rappresentazioni con i disegni sono molto più naturali e istruttive di qualsiasi altra alternativa. In questo capitolo vedremo le possibilità grafiche dell'Apple.

Il microelaboratore Apple II ha la possibilità di produrre sullo schermo, nel medesimo tempo, grafici e colori. Inoltre i grafici possono essere sia a «bassa risoluzione» che ad «alta risoluzione». Nel presente contesto la risoluzione si riferisce alla grandezza del singolo elemento grafico (pixel) che appare sullo schermo. I grafici a bassa risoluzione, così, si riferiscono a quelli in cui gli elementi sono molto grandi e la conseguente chiarezza e fedeltà di riproduzione dei disegni è

relativamente mediocre. I grafici ad alta risoluzione hanno elementi grafici molto più piccoli e la conseguente nitidezza e fedeltà dei disegni prodotti è maggiore.

I programmi che prevedono grafici, tendono ad usare soprattutto comandi a bassa risoluzione; vi raccomandiamo di far lo stesso, almeno fino a quando non siate diventati più esperti nella programmazione con l'Apple. Potrebbe sembrare che sia possibile fare soltanto una limitata gamma di disegni, nell'ambito di questo formato; ma, come hanno dimostrato molti programmi, e come mostrerà un certo numero di programmi in questo capitolo, è possibile produrre grafici di sorprendente complessità. Per produrli necessita pazienza e abilità, ma per cominciare c'è solo bisogno di una limitata conoscenza e di un po' di lavoro. Molta gente trova che escogitare ed usare le facilitazioni grafiche dell'Apple sia fra i suoi aspetti più interessanti. L'introduzione di buoni effetti grafici è stata certamente una delle maggiori ragioni del successo di molti dei migliori programmi scritti per microelaboratori; essa inoltre aiuta a garantire che i nuovi programmi diventino una fonte di piacere durevole ed utilità.

Lo schermo e la memoria

Un certo numero di microelaboratori usa un comando POKE per introdurre speciali simboli in particolari posizioni sullo schermo, al fine di elaborare disegni. Questo NON è il metodo usato dall'Apple; lo ricordiamo solo perché molti hanno sentito parlare di schermi a «mappe di memoria» e del comando POKE.

In effetti, il principio dei grafici, nell'Apple, è molto simile. Immaginate lo schermo come una griglia a piccoli quadretti. Ciascun quadretto può esser definito con coordinate X, Y. X è il numero del quadretto da sinistra a destra (cioè la colonna), Y è il numero dall'alto in basso (cioè la riga). Il quadretto a sinistra in alto è sempre 0,0.

Il totale dei quadretti attualmente dipende dalla risoluzione grafica che usate. Se usate la bassa risoluzione (fig. 4.1), è di 40 righe per 40 colonne.

COLONNE

	0	1	... C ...	38	39
0	0,0	1,0		38,0	39,0
1					
RIGHE R					
38					
39	0,39	1,39		38,39	39,39

Fig. 4.1 Mappa dello schermo a bassa risoluzione

Così nella griglia a bassa risoluzione, la prima riga di quadretti andrà da 0 a 39 da sinistra a destra e può essere indicata con 0,0; 1,0; 2,0; 3,0 fino a 39,0. Similmente l'ultima riga di quadretti può esser indicata dalle coordinate 0,39; 1,39; 2,39 fino a 39,39.

Lo schermo per grafici ad alta risoluzione è una griglia di quadretti di 280 per 160. Il quadretto in alto a sinistra è ancora 0,0; invece quello in alto a destra è ora 279,0, mentre il quadretto in basso a destra è 279,159.

Per usare la modalità grafica a bassa risoluzione, digitate:

GR<RETURN>

Questo prepara lo schermo per il grafico e lascia 4 righe per il testo, in basso.

L'Apple non possiede specifici caratteri grafici. Potete semplicemente «accendere» ognuno dei quadretti della griglia. Il comando per far ciò è PLOT X, Y. Potete, tuttavia, usare differenti colori che sono attivati dal comando:

COLOR = n

dove n è un numero compreso tra 0 e 15.

Prima di spiegare questi comandi, dovete ricordare 2 ulteriori istruzioni:

```
VLIN X,Y AT n
HLIN X,Y AT n
```

Naturalmente questi comandi tracciano sia righe verticali sia orizzontali. La riga verticale è tracciata sulla colonna n dal quadretto X fino a quello Y. La riga orizzontale è tracciata sulla riga n dal quadretto X a quello Y. Provate quanto detto, digitando la seguente sequenza nella modalità IMMEDIATA:

```
GR<RETURN>
COLOR=1<RETURN>
HLIN 0,30 AT 20<RETURN>
VLIN 0,39 AT 20<RETURN>
```

(Notate che abbiamo posto il COLOR=1. È necessario poiché il COLOR assegnato automaticamente, cioè lo zero, è lo stesso COLOR di sfondo dello schermo. In altri termini, se lasciate lo schermo al COLOR=0, il programma lavorerà, ma voi non lo potrete vedere). In relazione al tipo di video che usate, potrete vedere ora una debolissima croce sullo schermo. La maggior parte usa un Apple con uno schermo bianco su nero, o verde su nero. Questo è un video monocromatico. Alcuni lettori, specialmente coloro che hanno un'interfaccia per la televisione, sono in grado di usare grafici colorati. Vi consigliamo di sperimentare il colore che meglio appare sul vostro video. Potrebbe essere «15».

Digitate questo breve programma per scoprirlo:

```
10 GR
20 FOR I=0 TO 15
30 COLOR=I
40 VLIN 0,39 AT I
50 NEXT I
60 PRINT "0123456789111111"
70 PRINT "(SPC x 10)012345"
```

Per usare tutti i comandi disponibili, ed anche per poter vedere quale colore sia il migliore, fate le seguenti modifiche

al programma. Cancellate le righe 60 e 70. Aggiungete le seguenti:

```
100 FOR K=0 TO 15
110 COLOR=K
120 PRINT
130 PRINT "0123456789111111(SPC×5) È IL COLO
RE (SPC)";K
140 PRINT "(SPC×10)012345"
150 FOR I=20 TO 39
160 FOR J=0 TO 39
170 PLOT I,J
180 NEXT J: NEXT I: NEXT K.
```

Notate che il comando per tornare dallo schermo grafico allo schermo per i testi è:

TEXT<RETURN>

Fare un disegno

Un'«invasore spaziale» è un'immagine fittizia nel senso che prende forma dai caratteri grafici piuttosto che un intrinseco aspetto di ciò che è modellato o approssimato, usando i caratteri grafici. In questa parte viene descritta una procedura per abbozzare un'immagine sullo schermo. Ciò dimostra che si possono produrre schizzi riconoscibili mentre, allo stesso tempo, si dimostra che la definizione limitata di bassa risoluzione può dare problemi sull'accuratezza dello schizzo.

Supponete che si voglia disegnare sullo schermo la farfalla della fig. 4.2 (a). Per far ciò, disegnateci sopra una griglia come mostra la fig. 4.2 (b) e poi, tutto intorno, prendete il quadretto della griglia più vicino al disegno. Risultato di ciò sarà qualcosa simile a quanto mostra la fig. 4.2 (c), mentre il contorno della farfalla, come apparirà sullo schermo, è mostrato dalla fig. 4.2 (d). Infine trovate le coordinate dei quadretti sullo schermo e scrivete un programma per porli al posto giusto sullo schermo. Il miglior metodo è di usare i comandi BASIC READ e DATA. Il comando READ legge i dati da una lista DATA. Il primo comando eseguito in un

programma leggerà il primo dato dalla lista DATA, il secondo prenderà il secondo dato e così via. Ponendo il comando READ in un loop FOR...NEXT, si potrà facilmente leggere l'intera lista DATA.

La struttura del seguente programma della farfalla è semplice. La prima riga contiene due comandi; pone Y sulla prima riga del disegno sullo schermo (= 15) e sceglie la modalità grafica. Il resto del programma è diviso in due parti ben distinte: la prima è un complesso loop FOR...NEXT che contiene un secondo loop (FOR J). È chiamato «nidificato». La seconda parte del programma (righe 300-360) contiene i dati che fanno posizionare i componenti grafici sullo schermo per disegnare la farfalla.

```
 5 HOME
10 Y = 15: GR
20 FOR I = 1 TO 16
30 READ N: N = INT (N/2)
40 X1 = 1
50 FOR J = 1 TO N
60 COLOR = 0
```

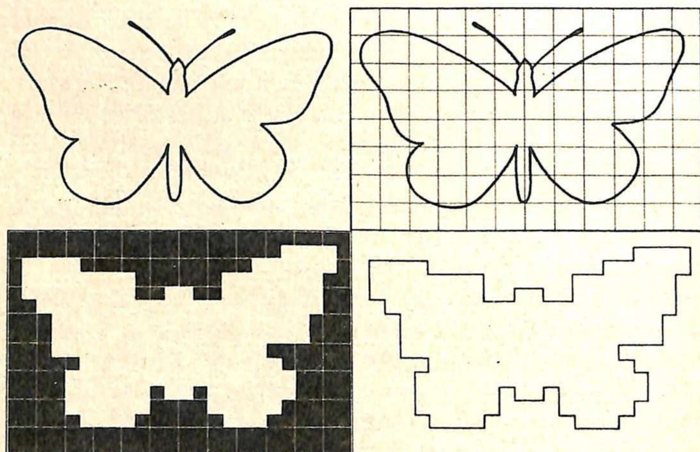


Fig. 4.2 (a) Farfalla. (b) Farfalla sulla griglia. (c) Farfalla formata da elementi grafici. (d) Contorno del disegno.

```
 70 READ X2
 80 HLIN X1,X2 AT Y
 90 COLOR = 3
100 X1 = X2 + 1
110 READ X2
120 HLIN X1,X2, AT Y
130 NEXT J
140 COLOR = 0
150 X1 = X2 + 1
160 HLIN X1,15 AT Y
170 Y = Y + 1
180 NEXT I
300 DATA 3,10,12,5,3,5,10,13
310 DATA 5,1,5,10,14,7,1,6,7,8,9,14
320 DATA 3,1,14,3,1,14,3,1,14
330 DATA 3,1,14,3,2,13,3,3,12
340 DATA 7,1,6,7,8,9,14,7,1,6,7,8,9,14
350 DATA 7,1,6,7,8,9,14,5,1,6,9,14
360 DATA 5,3,5,10,13,5,3,5,11,13
400 GOTO 400
```

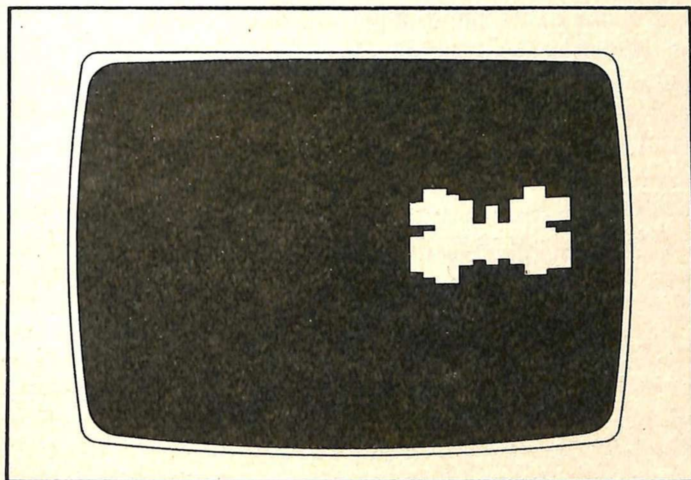


Fig. 4.3 Farfalla disegnata sullo schermo

La riga 400 evita soltanto che il programma termini e che appaia il «pronto»; evita inoltre che il cursore rovini il disegno sullo schermo. Il programma può esser fermato premendo i tasti <CTRL>/C (oppure <CTRL>/<RESET>). Se scegliete l'ultimo sistema, non solo fermerete il programma ma uscirete anche dalla modalità grafica per tornare alla modalità testo. Il risultato sarà un insieme di caratteri che ricoprono tutti quelli usati per il disegno sullo schermo. Digitando semplicemente:

```
HOME<RETURN>
```

si pulisce lo schermo. Se spegnete l'elaboratore, non solo fermate il programma, ma lo cancellate dalla memoria.

La figura prodotta dal programma è mostrata dalla fig. 4.3. Ciascun problema di risoluzione può esser affrontato in diversi modi: il più semplice è di allontanarsi dallo schermo, lasciando che gli occhi ed il cervello integrino e risolvano l'immagine finché il dettaglio minuto diventi chiaro. Un metodo migliore è di scegliere la modalità grafica ad alta risoluzione che rende possibile l'uso di un gran numero di quadretti della griglia. (Tratteremo i grafici ad alta risoluzione più avanti, in questo capitolo).

È anche molto importante posizionare la griglia poiché possono esser presi con maggiore accuratezza i dettagli che sono fondamentali per il riconoscimento. Infine, una piccola licenza artistica può esser anche un aiuto considerevole nel mostrare i disegni.

Modelli di schermo

Lo schermo può esser riempito con un simbolo con il programma:

```
10 HOME
20 FOR I = 1TO 40
30 FOR J = 1 TO 24
40 HTAB I
50 VTAB J
60 PRINT «*»
70 NEXT J,I
```

```
80 GOTO 80
```

Usando la modalità grafica, possiamo riempire lo schermo con un determinato colore piuttosto che con un carattere.

```
10 HOME
20 GR
25 COLOR=1
30 FOR I=0 TO 39: FOR J=0 TO 39
40 PLOT I,J
50 NEXT J: NEXT I
60 GOTO 60
```

Aggiungendo un loop FOR...NEXT, possiamo riempire lo schermo con colori diversi, uno dopo l'altro:

```
10 HOME:GR
20 FOR K=1 TO 15
30 FOR I=0 TO 39
40 FOR J=0 TO 39
50 COLOR=K
60 PLOT I,J
70 NEXT J,I,K
80 GOTO 80
```

Per comodità e velocità, abbiamo usato NEXT in una sola riga utilizzando le virgole per separare i comandi. Quando un metodo preciso ed una funzione producono le posizioni dei caratteri sullo schermo, si avrà sia una informazione sia un'estetica gradevole. Uno schema generale che può essere usato per creare un'ampia varietà di modelli, coinvolge i tre livelli di calcolo, clasificazione e rappresentazione. Il valore è calcolato per ogni posizione dello schermo, usando i numeri di riga e colonna. Il valore è anche classificato assegnandogli un numero di classe. Ogni classe è rappresentata da un colore. Si può ottenere un carattere per ogni posizione dello schermo e disegnarlo. La metodologia consiste nel fare mappe colorate dove l'altezza di ogni punto è misurata e calcolata, classificata secondo l'altezza e rappresentata nella mappa del colore assegnato a quell'intervallo di altezza. Uno schema generale di programma per produrre sullo

schermo modelli di questo tipo è in fig. 4.4; e abbiamo anche un tipo di programma:

```
 10 HOME
 20 FOR R=1 TO 24
 30 FOR C=1 TO 40
 35 HTAB C: VTAB R
 40 H=R*R+C*C
 50 IF H<60 THEN I=35
 60 IF H>60 THEN I=43
 70 IF H>200 THEN I=36
 80 IF H>400 THEN I=58
 90 PRINT CHR$(I);
100 NEXT C,R
110 GOTO 110
```

Se ordinate con cura il programma, noterete che segue esattamente lo schema di programma di Fig. 4.4. La figura 4.5 mostra il risultato sullo schermo. La riga 20 è un FOR per le righe, la riga 30 è un FOR per le colonne. La riga 40 calcola il valore che le righe da 50 a 80 classificano. la riga 90 stampa il carattere individuato in un punto dello schermo. Il tipo di simbolo da stampare è definito dal comando BASIC CHR$(n), dove n è il codice che identifica i diversi caratteri.

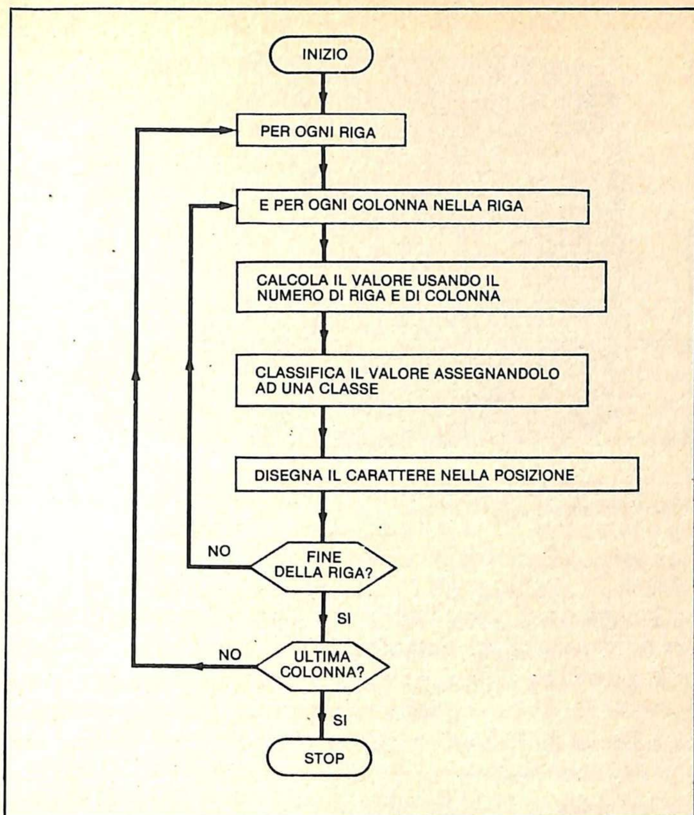


Fig. 4.4 Diagramma di flusso

Per conoscere tutti i caratteri disponibili con questo comando, provate il seguente programma:

```
10 HOME
20 FOR I = 32 TO 127
30 PRINT "IL CODICE DI (SPC)";CHR$(I);"(SPC) È
(SPC)"; I
40 FOR J = 0 TO 400:NEXT J
50 HOME
60 NEXT I
```

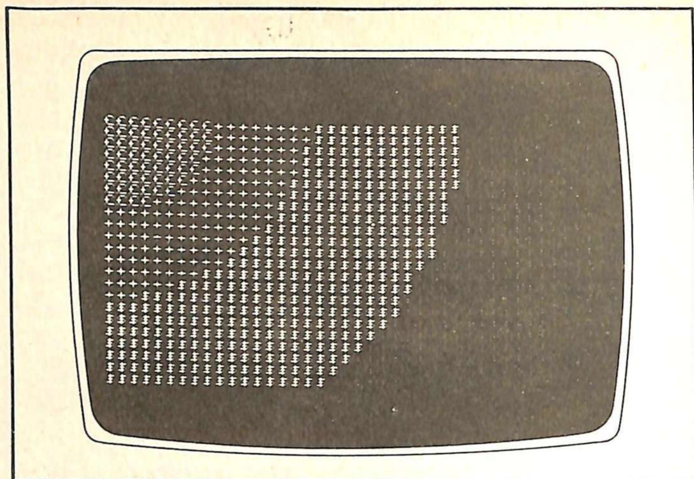


Fig. 4.5 Disegno sullo schermo

Il risultato dipende dal tipo di Apple che usate. L'apparente mancanza del simbolo CHR$(32) è causata dal fatto che il carattere è uno spazio. Se i caratteri lampeggiano attraverso lo schermo troppo velocemente, potete o aumentare il valore 400 alla riga 40, portandolo per esempio a 600; oppure potete cancellare la riga 50 e vedere cosa succede. La riga 40 è un «loop di ritardo». È un loop FOR...NEXT senza alcun'altra istruzione intermedia. Cioè, questo comando ordina all'elaboratore di non fare niente per 400 volte (eccetto che girare su se stesso): così, quanto più grande è il numero, tanto più lungo è il ritardo.

Ritornando al nostro modello, un programma che segue lo stesso schema è:

```
10 HOME
20 DIM A(10)
30 FOR I=1 TO 10
40 READ A(I)
50 NEXT I
60 DATA 35,36,37,38,42,43,61,63,92,94
```

```
70 FOR R=1 TO 24
80 FOR C=1 TO 40
85 HTAB C: VTAB R
90 H=(R*C)∧(1/3)
100 H=INT(H)+1
110 PRINT CHR$(A(H));
120 NEXT C: NEXT R
130 GOTO 130
```

Qui ci sono 10 intervalli e simboli da scrivere. La riga 90 calcola il valore (R*C elevato alla potenza di 1/3, cioè la radice cubica di R*C), mentre la riga 100 classifica il valore prendendone la parte intera (il comando INT). Per cambiare l'aspetto del modello, provate a cambiare il valore dei simboli da usare. Per esempio potreste cambiare tutti i valori nella lista DATA.

Con questo metodo si possono produrre molti modelli. In generale un modello è il risultato della scelta del metodo di calcolo, di classificazione e dei simboli che rappresentano le classi. La classificazione può esser effettuata non solo dividendo in intervalli il campo dei valori, ma anche in altri modi; per esempio la cifra nel primo posto dopo la virgola del valore calcolato può esser usata per assegnare la classe. La scelta dei caratteri da stampare è molto importante per la presentazione dei modelli. I caratteri scelti negli ultimi due programmi fanno risaltare il passaggio da una classe all'altra, ma altri caratteri possono dare un effetto migliore.

Movimenti

Una volta fatti dei disegni statici, è naturale passare ad elaborare disegni in movimento. I programmi di questo paragrafo vi permetteranno di controllare il movimento di una figura sullo schermo. Oltre ad essere attraenti, tali programmi illustrano le tecniche usate in molti video-giochi.

È necessario ricordare che l'Apple può usare joysticks per controllare il movimento sullo schermo; ma, essendo una tecnica di programmazione molto avanzata, non sarà trattata in questo libro.

Usando le istruzioni necessarie a disegnare un invasore spaziale, potrete scrivere un programma che muova l'invasore avanti e indietro attraverso lo schermo. Il programma controlla la tastiera per vedere se sia stato spinto qualche tasto di «movimento» e, in tal caso, spostare di conseguenza l'invasore. È necessaria una tecnica raffinata per evitare che la figura, muovendosi, lasci la propria traccia. Si simula il movimento ridisegnando, a sinistra o a destra di una posizione, l'intera figura (v. figura 4.6). Se si muove verso destra, lascerà a sinistra alcuni caratteri nel punto in cui era sullo schermo. Un mezzo per evitare questo è di avere intorno alla figura un contorno di spazi, così la parte·indietro a sinistra è sempre formata da spazi.

```
  4 HOME
  5 GR: COLOR = 3
 10 DIM X(60),Y(60)
 20 FOR I = 1 TO 31: READ X(1): NEXT I
 30 DATA 3,4,5,2,3,4,5,6,1,2,4,6,7,1,2,3,4,5,6,7,2,3,5,6,
3,4,5,2,6,1,7
 40 FOR I = 1 TO 31:READ Y(I):NEXT I
 50 DATA 1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,
6,6,6,7,7,8,8
 60 FOR I = 32 TO 54: READ X(I): NEXT I
 70 DATA 2,6,1,7,0,3,5,8,0,8,1,4,7,2,6,1,3,5,7,0,2,6,8
 80 FOR I = 32 TO 54: READ Y(I): NEXT I
 90 DATA 1,1,2,2,3,3,3,3,4,4,5,5,5,6,6,7,7,7,7,8,8,8,8
100 X = 10: Y = 10
110 COLOR = 3
120 FOR I = 1 TO 31: PLOT X + X(I),Y + Y(I): NEXT I
130 GET A$: IF A$ = "(SPC)" THEN 130
140 COLOR = 0
150 IF ASC(A$)<>8 AND ASC(A$)<>21 THEN
GOTO 130
160 IF ASC(A$) = 8 THEN X = X—1
170 IF ASC(A$) = 21 THEN X = X + 1
180 IF X<1 OR X>30 THEN GOTO 130
```

```
190 FOR I = 32 TO 54: PLOT X + X(I), Y + Y(I): NEXT I
200 GOTO 110
```

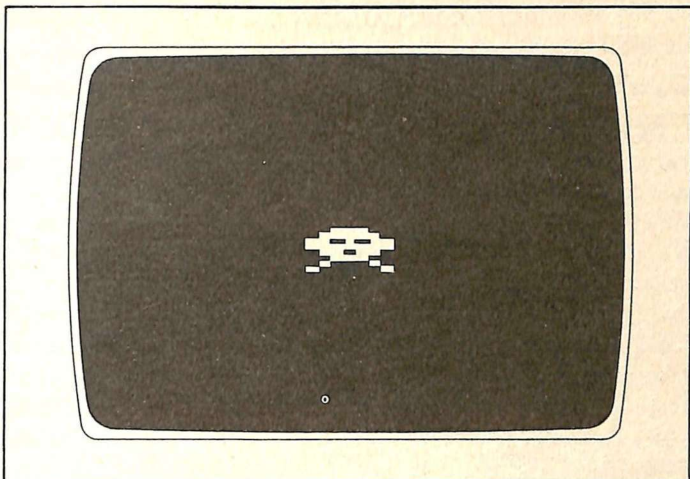


Fig. 4.6 Il movimento di un invasore spaziale

Per esser sicuri che seguiate ciò che fa il programma, vi consigliamo di fare così: per prima cosa digitate:

SPEED<RETURN>

Questa istruzione fa scendere la velocità con cui il programma disegna le figure fino a rendere quasi visibili le operazioni che compie. In questo modo potete vedere ogni singola azione del programma. Ora digitate:

TRACE<RETURN>

Questo comando fa stampare tutti i numeri di riga dei comandi mentre si esegue il programma. Poiché la velocità è molto bassa, potrete vedere chiaramente l'effetto di ogni comando. Ogni volta che abbiate necessità di fermare il programma, per comprenderne la logica, digitate:

<CTRL>/C

L'Apple emetterà un bip e visualizzerà il messaggio:

BREAK IN n

dove n è il numero di riga che si stava eseguendo al momento dell'interruzione. Ciò fermerà il programma. Quando siete pronti a continuare, digitate:

CONT<RETURN>

Se fate ciò al programma precedente, vedrete che l'elaboratore impiega molto tempo per leggere i dati nelle istruzioni alle righe 20, 40, 60, 80 e memorizzarli. Queste sono le coordinate X e Y che faranno disegnare alla riga 120 la figura dell'invasore spaziale. Quando la figura sarà disegnata sullo schermo, il programma si fermerà alla riga 130. Se guardiamo la lista del programma, troviamo un nuovo comando: GET A$. Questo è simile all'istruzione INPUT, ma non vi chiede di spingere <RETURN>. In altre parole, GET A$ attende che sia digitato sulla tastiera un carattere e lo assegna subito alla variabile scritta dopo il comando; in questo caso A$.

Non appena il tasto viene spinto, il programma continua. La riga 150 controlla se il tasto (che è memorizzato in A$), sia una freccia a sinistra o a destra. Se non è nessuno dei due casi, il programma torna alla riga 130 dove aspetta che sia spinto un altro tasto. A questo punto non si può usare <CTRL>/C poiché non è né una freccia a sinistra né a destra. Provate. Se, tuttavia, il valore di A$ è 8 (sinistra), o 22 (destra), allora il programma aggiunge o sottrae 1 dalla X (coordinata di colonna). La riga 190 è usata per disegnare spazi e cancellare la traccia. Infine la riga 200 fa tornare il programma alla riga 110 per disegnare la nuova posizione.

Osserviamo la riga 180. È usata per indicare il bordo sinistro e destro dello schermo e impedisce all'invasore di uscire dallo schermo. Quando avete finalmente terminato di esaminare il programma, digitate:

NOTRACE<RETURN>
SPEED=255<RETURN>

Animazione

Visualizzare disegni ad una frequenza sufficiente produce l'illusione del movimento. Tutti i sistemi di disegni animati, i films e la televisione usano questo effetto che dipende dal fenomeno chiamato «persistenza dell'immagine». Il programma presentato in questo paragrafo produce un disegno animato disegnando una serie di disegni statici nello stesso modo con cui un disegno animato è l'effetto della visione a velocità sufficiente di disegni statici.

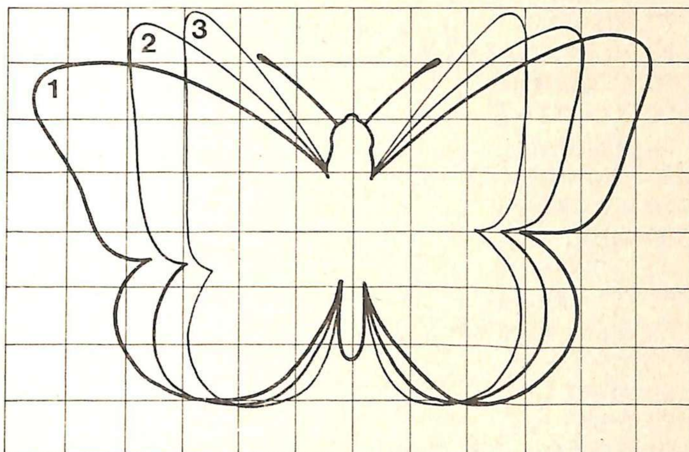


Fig. 4.7 Immagini 1, 2 e 3 della farfalla in volo

Il seguente programma produce il disegno animato di una farfalla in volo: le immagini successive danno la forma diversa della farfalla mentre vola. La successione di movimenti da cui sono derivate le immagini è mostrata nella fig. 4.7. La prima immagine, con le ali completamente aperte, è l'immagine prodotta nei precedenti paragrafi. Le altre immagini sono ottenute come la prima. Nel programma vengono prima letti i codici per formare le tre immagini e poi le immagini sono ripetutamente disegnate con la sequenza 1,2,3,2,1. Il

diagramma di flusso del programma è in fig. 4.8. Il programma è:

```
  4 HOME
  5 GR
 10 FOR K = 1 TO 4
 15 Y = 15
 20 FOR I = 1 TO 16
 30 READ N: N = INT(N/2)
 40 X1 = 1
 50 FOR J = 1 TO N
 60 COLOR = 0
 70 READ X2
 80 HLIN X1,X2 AT Y
 90 COLOR = 3
100 X1 = X2 + 1
110 READ X2
120 HLIN X1,X2 AT Y
125 X1 = X2 + 1
130 NEXT J
140 COLOR = 0
150 X1 = X2 + 1
160 HLIN X1, 15 AT Y
170 Y = Y + 1
180 NEXT I
190 NEXT K
200 RUN 10
300 DATA 3,10,12,5,3,5,10,13
310 DATA 5,1,5,10,14,7,1,6,7,8,9,14
320 DATA 3,1,14,3,1,14,3,1,14
330 DATA 3,1,14,3,2,13,3,3,12
340 DATA 7,1,6,7,8,9,14,7,1,6,7,8,9,14
350 DATA 7,1,6,7,8,9,14,5,1,6,9,14
360 DATA 5,3,5,10,13,5,3,5,11,13
370 REM
380 DATA 5,4,5,10,11,5,3,5,10,12
390 DATA 5,2,5,10,13,7,2,6,7,8,9,13
400 DATA 3,2,13,3,2,13,3,2,13,3,2,13
410 DATA 3,3,12,3,4,12,7,2,6,7,8,9,13
```

```
420 DATA 5,2,6,9,13,5,2,6,9,13,5,2,6,9,13
430 DATA 5,3,5,10,12,5,3,5,11,12
440 REM
450 DATA 5,4,5,10,11,5,3,5,10,12
460 DATA 5,3,5,10,12,7,3,6,7,8,9,12
470 DATA 3,3,12,3,3,12,3,3,12,3,3,12,3,4,11,3,3,12
480 DATA 7,3,6,7,8,9,12,5,3,6,9,12,5,3,6,9,12
490 DATA 5,3,6,9,12,5,4,5,10,11,5,4,5,10,11
500 REM
510 DATA 5,4,5,10,11,5,3,5,10,12
520 DATA 5,2,5,10,13,7,2,6,7,8,9,13
530 DATA 3,2,13,3,2,13,3,2,13,3,2,13
540 DATA 3,3,12,3,4,12,7,2,6,7,8,9,13
550 DATA 5,2,6,9,13,5,2,6,9,13,5,2,6,9,13
560 DATA 5,3,5,10,12,5,3,5,11,12
```

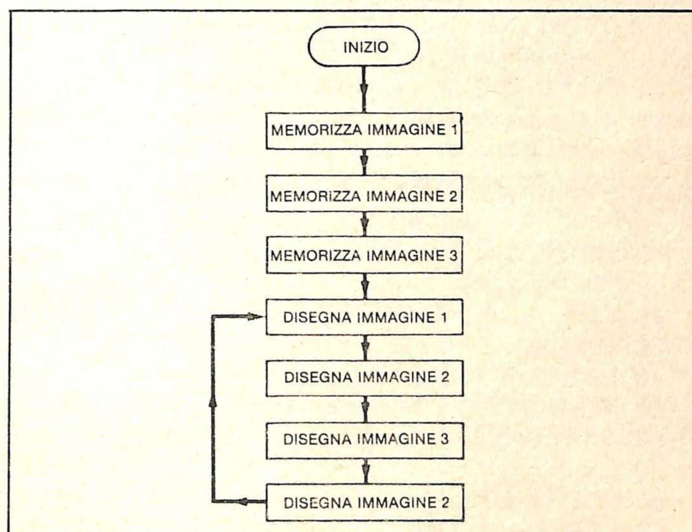


Fig. 4.8 Diagramma di flusso del disegno animato

Questo programma ha dimostrato come si ottenga la simulazione di animazione. Non può lavorare ad una velocità tale da essere eguale alla realtà. Ovviamente più le immagini

sono grandi e complicate e più tempo usa l'elaboratore per mostrare le diverse immagini e così diminuisce la capacità di animazione. Un metodo da usare per aumentare la frequenza dell'animazione è disegnare solo i cambiamenti necessari per passare da un'immagine alla successiva senza ridisegnare tutta l'immagine. Potete anche fare l'animazione di un disegno molto piccolo per accorgervi quanto si muova più velocemente e più aderentemente alla realtà. Se avete il programma di dimostrazione Applevision, fatelo girare per vedere un esempio reale di animazione.

Simulazione dinamica

Questo paragrafo effettua una simulazione dinamica di un sistema che casualmente cresce e diminuisce. Visualizza una comunità che inizialmente cresce di un solo elemento alla volta. Quando raggiunge una determinata grandezza, decresce sino ad un livello più basso e fluttua tra questi due livelli. Si può prendere per la simulazione la crescita di una città, di una colonia di insetti, potrete anche capire che i piani per la crescita delle città, per le vere città, non crescono casualmente! Il numero casuale necessario per il programma è generato dal comando RND che genera un numero pseudo-casuale.

```
  5 HOME: G = 1: C = 1
 10 DIM A(12,16)
 20 T = 1: A(5,6) = 1: GOSUB 1000
 30 FOR I = 1 TO 12
 40 FOR J = 1 TO 16
 50 IF RND(1)>0.9 THEN A(I,J) = T
 60 NEXT J: NEXT I
 70 C = 0
 80 FOR I = 1 TO 12
 90 FOR J = 1 TO 16
100 IF A(I,J) = 1 THEN C = C + 1
110 NEXT J: NEXT I
120 G = G + 1: GOSUB 1000
130 IF C>99 THEN T = 0
140 IF C<27 THEN T = 1
```

```
150 GOTO 30
1000 FOR I = 1 TO 12
1010 FOR J = 1 TO 16
1020 IF A(I,J) = 1 THEN VTAB (I): HTAB(J): PRINT"*"
1030 IF A(I,J) = 0 THEN VTAB (I): HTAB(J): PRINT
"(SPC)"
1040 NEXT J: NEXT I
1050 HTAB(1): VTAB(22): PRINT"GENERAZIONE
(SPC)" G: PRINT: PRINT"POPOLAZIONE(SPC)";C;
"(SPC)"
1060 RETURN
```

Grafica ad alta risoluzione

Per molti microelaboratori, la grafica ad alta risoluzione è così complicata da usare solo grafica a bassa risoluzione. Con l'Apple II, invece, è molto semplice. Ricordiamo che per l'alta risoluzione la grandezza dello schermo è di 280 colonne per 160 righe, e che il comando per passare a tale schermo è

```
HGR<RETURN>
```

Vi sono solo due comandi che è necessario ricordare per fare semplici disegni ad alta risoluzione:

```
HCOLOR
HPLOT
```

L'uso di HCOLOR è ovvio e dovrete solo provare per trovare i colori disponibili e che preferite usare. HPLOT ha due forme: HPLOT X,Y e HPLOT X1,Y1 TO X2,Y2 con X e Y che rappresentano rispettivamente le colonne e le righe. Digitate:

```
HGR<RETURN>
HCOLOR = 3<RETURN>
HPLOT 140,80<RETURN>
```

Vedrete un piccolo punto al centro dello schermo. Ora digitate:

```
HPLOT TO 279,0<RETURN>
```

Vedrete disegnare una linea dall'ultima posizione alla nuova posizione. Infine digitate:

HPLOT 0,0 TO 140,80<RETURN>

Questa volta vedrete una linea andare dall'angolo in alto a sinistra al centro dello schermo. Per dimostrare le possibilità grafiche dell'Apple, provate il seguente programma. È abbastanza lungo, ma la logica è semplicissima ed usa solo i comandi che avete appena imparato ad usare. Il programma domanda il punto dello schermo da dove volete che inizi la linea. Chiede anche la direzione in cui volete disegnare la linea (N,S,E,O,NE,SO...) e la lunghezza che desiderate.

Le righe 100 e 180 richiedono differenti routine che dipendono dalla risposta; ciascuna di esse contiene una sottoroutine per impedire che i valori siano troppo alti. Digitate:

HPLOT 0,80 TO 280,80<RETURN>

per vedere perché ciò sia necessario. Il programma, che possiamo chiamare «LAVAGNA PER DISEGNI» (v. fig. 4,9) è il seguente:

```
 10 HGR
 11 PRINT
 12 PRINT "DA QUALE PUNTO INIZIA?"
 13 PRIMT "USA IL FORMATO: X,Y"
 14 PRINT "X=COLONNA, Y=RIGA";
 15 INPUT X,Y
 16 HCOLOR=3
 17 IF X>279 THEN X=279
 18 IF Y>159 THEN Y=150
 20 P=0
 30 PRINT "QUALE DIREZIONE? USA I PUNTI CAR
DINALI"
 35 PRINT
 40 PRINT "QUALE LUNGHEZZA? USA IL FORMA
TO=N,10"
 60 INPUT DIR$,SIZ
 70 P=P+1
100 IF DIR$="N" THEN GOTO 200
110 IF DIR$="S" THEN GOTO 300
```

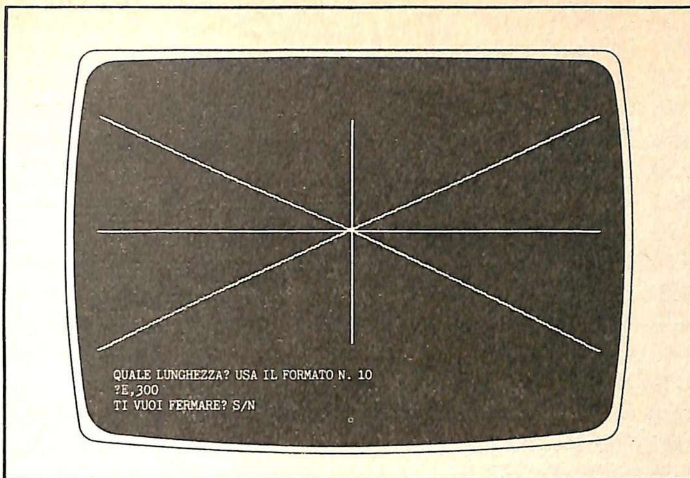


Fig. 4.9 Visualizzazione «lavagna per disegni»

```
120 IF DIR$ = "O" THEN GOTO 500
130 IF DIR$ = "E" THEN GOTO 400
140 IF DIR$ = "NE" THEN GOTO 600
150 IF DIR$ = "NO" THEN GOTO 700
160 IF DIR$ = "SE" THEN GOTO 800
170 IF DIR$ = "SO" THEN GOTO 900
180 PRINT "INPUT NON CORRETTO. RIPROVA"
190 FOR I = 1 TO 750: NEXT I: GOTO 30
199 REM
200 REM       DIREZIONE NORD
201 REM
205 SIZ = Y-SIZ
210 GOSUB 5200
220 HPLOT X,Y TO X,SIZ
230 Y = SIZ
240 GOTO 1000
299 REM
300 REM       DIREZIONE SUD
301 REM
305 SIZ = Y + SIZ
```

```
310 GOSUB 5300
320 HPLOT X,Y TO X,SIZ
330 Y = SIZ
340 GOTO 1000
399 REM
400 REM         DIREZIONE EST
401 REM
405 SIZ = X + SIZ
410 GOSUB 5400
420 HPLOT X,Y TO SIZ,Y
430 X = SIZ
440 GOTO 1000
499 REM
500 REM         DIREZIONE OVEST
501 REM
505 SIZ = X-SIZ
510 GOSUB 5200
520 HPLOT X,Y TO SIZ,Y
530 X = SIZ
540 GOTO 1000
599 REM
600 REM         DIREZIONE NORD/EST
601 REM
605 XSIZ = X + SIZ: YSIZ = Y-SIZ
610 GOSUB 5600
620 HPLOT X,Y TO XSIZ,YSIZ
630 X = XSIZ: Y = YSIZ
640 GOTO 1000
699 REM
700 REM         DIREZIONE NORD/OVEST
701 REM
705 XSIZ = X-SIZ: YSIZ = Y-SIZ
710 GOSUB 5700
720 HPLOT X,Y TO XSIZ,YSIZ
730 X = XSIZ: Y = YSIZ
740 GOTO 1000
799 REM
800 REM         DIREZIONE SUD/EST
801 REM
```

```
805 XSIZ = X + SIZ: YSIZ = Y + SIZ
810 GOSUB 5800
820 HPLOT X,Y TO XSIZ,YSIZ
830 X = XSIZ: Y = YSIZ
840 GOTO 1000
899 REM
900 REM      DIREZIONE SUD/OVEST
901 REM
905 XSIZ = X-SIZ: YSIZ = Y + SIZ
910 GOSUB 5900
920 HPLOT X,Y TO XSIZ,YSIZ
930 X = XSIZ: Y = YSIZ
940 GOTO 1000
1000 PRINT "TI VUOI FERMARE? S/N"
1010 INPUT QA$: IF QA$ = "S" THEN GOTO 6000
1020 PRINT "INIZIA DA UN NUOVO PUNTO?"
1030 INPUT QB$: IF QB$ = "S" THEN GOTO 12
1040 PRINT "CONTINUA DA QUESTO PUNTO? S/N"
1050 INPUT QC$: IF QC$ = "S" THEN GOTO 30
1060 PRINT "INPUT NON CORRETTO. RIPROVA"
1070 FOR I = 1 TO 750: NEXT I: GOTO 1000
5200 IF SIZ<0 THEN SIZ = 0
5210 RETURN
5300 IF SIZ>159 THEN SIZ = 159
5310 RETURN
5400 IF SIZ>279 THEN SIZ = 279
5410 RETURN
5600 IF XSIZ>279 THEN XSIZ = 279
5610 IF YSIZ<0 THEN YSIZ = 0
5620 RETURN
5700 IF XSIZ<0 THEN XSIZ = 0
5710 IF YSIZ<0 THEN YSIZ = 0
5720 RETURN
5800 IF XSIZ>279 THEN XSIZ = 279
5810 IF YSIZ>159 THEN YSIZ = 159
5820 RETURN
5900 IF XSIZ<0 THEN XSIZ = 0
5910 IF YSIZ>159 THEN YSIZ = 159
6000 IF P>10 THEN GOTO 6500
```

```
6008 PRINT
6009 PRINT
6010 PRINT "QUESTO È IL RISULTATO"
6020 FOR I = 1 TO 1000: NEXT I
6028 PRINT
6029 PRINT
6030 PRINT "DEVI AVERE PIÙ IMMAGINAZIONE"
6040 FOR I = 1 TO 1000: NEXT I
6048 PRINT
6049 PRINT
6050 PRINT "PENSACI"
6060 FOR I = 1 TO 1000: NEXT I
6500 PRINT
6510 PRINT
6520 PRINT "CIAO"
6530 END
```

Potreste prendere questo programma come base e compilarlo migliorandolo.

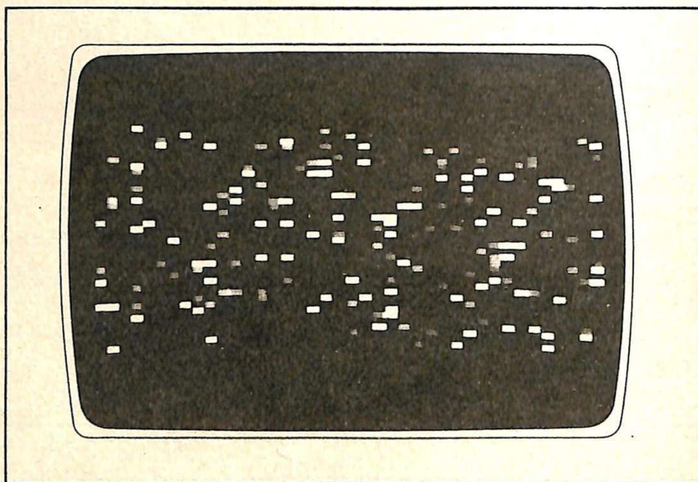


Fig. 4.10 Programma «Colori casuali»

Autotest

Nel corso di questo capitolo abbiamo parlato di grafici ad alta e bassa risoluzione. Abbiamo visto come le formule dei colori possano venir usate con grafici a bassa risoluzione per disegnare modelli sullo schermo. Il seguente breve programma genera la produzione di differenti colori sullo schermo (v.f. 4.10).

```
 5 HOME
10 GR
20 V = INT(RND(1)*40): H = INT(RND(1)*40)
30 COLOR = INT(RND(1)*16)
40 PLOT V,H
50 GOTO 20
```

Adattate questo programma, usando il metodo del calcolo, della classificazione e della rappresentazione, per produrre simmetricamente i colori, come fa il caleidoscopio.

Capitolo 5

Caratteristiche dell'Apple ed accessori

In questo capitolo diamo informazioni sull'Apple e sui suoi accessori che sono stati menzionati solo brevemente nei precedenti capitoli. Non intendiamo dare un'informazione completa delle caratteristiche e degli accessori dell'Apple ma solo di quelle che possono essere utili ad un nuovo utilizzatore dell'elaboratore.

Caratteristiche dell'Apple

Costruttore:	Apple Computers Inc.
Microprocessore:	Synertek/MOS Technology 6502
Schermo:	40 caratteri, 24 linee

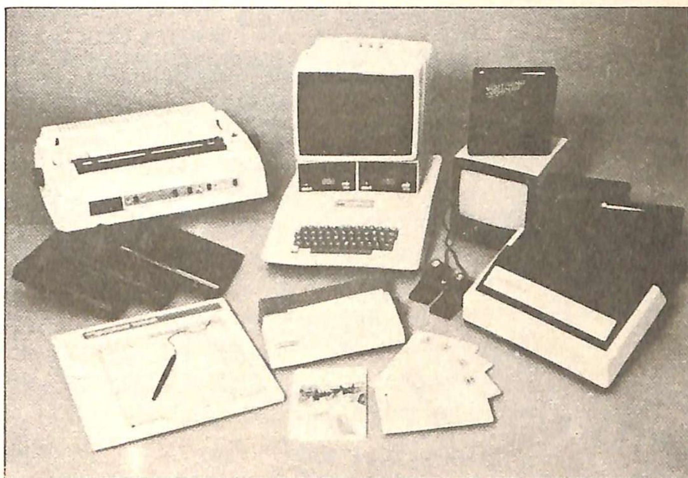


Fig. 5.1 L'Apple II con alcune periferiche

Tastiera:	52 tasti più barra spaziatrice tipo QWERTY
Memoria:	Da 4K a 48K. Le dimensioni più usuali sono 16K, 32K e 48K
Linguaggi:	Apple Integer BASIC, Applesoft BASIC
Grafica:	Bassa risoluzione — 16 colori a griglia 40×40 Alta risoluzione — due griglie: 280×160; 280×192 Colori: i colori disponibili dipendono dalla modalità.
Periferiche:	L'Apple II si può collegare ad un normale televisore a colori (oppure bianco e nero) con un adattatore video oppure ad un registratore a cassetta.

Drive

Il sottosistema Apple Disk II aumenta la capacità dell'Apple usando i dischetti per memorizzare i dati. Si può così aumentare la capacità di memoria, aumentare la velocità di ricerca dei dati e si può accedere casualmente ai dati memorizzati. Usare un Apple con il sistema Disk II negli affari, a scuola o in casa è la miglior risposta al problema della memorizzazione e della ricerca dei dati.

Stampante Termica Silentype

La stampante Silentype è una piccola stampante termica silenziosa e compatta. La Silentype riceve dall'elaboratore sia l'alimentazione sia «l'intelligenza» e potete programmare la stampante quando programmate l'Apple. La si usa per stampare testi e grafici. È silenziosa, per lavorare ovunque ed è leggera.

Tavoletta grafica

La tavoletta grafica rende l'Apple uno strumento artistico o un disegnatore di

immagini, giacché la tavoletta è il mezzo per fare e visualizzare disegni elettronici. Alcune applicazioni sono i diagrammi a blocco, gli schemi, i disegni di parti meccaniche.

È lo strumento ideale per trasformare le idee in realtà visibile.

Joystick

Il Joystick dell'Apple è uno strumento di facile uso per ogni genere di programmi. Può controllare i movimenti a 360 gradi ed ha un bottone per «far fuoco».

Espansione:

Cartolina di Interfaccia

La Cartolina di Interfaccia Seriale permette all'Apple di scambiare i dati con altri elaboratori, con la stampante e con altri accessori standard. Tale cartolina è facilmente controllabile dal BASIC.

La cartolina di interfaccia per la stampante sia Parallela sia Centronics vi permette di stampare scritti, listati, indirizzi e lettere, usando molti tipi di stampanti.

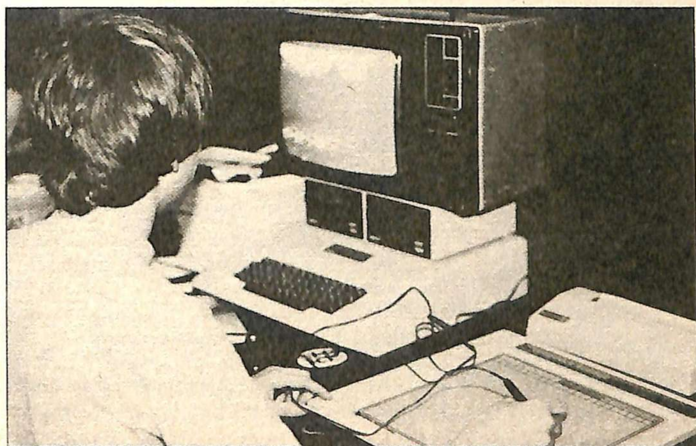


Fig. 5.2 La «Tavoletta grafica» dell'Apple

Cartoline Opzionali

Apple Language System: sostituisce automaticamente il BASIC in ROM con il linguaggio di programmazione Pascal contenuto in una RAM di 16K.

Applesoft Firmware Card: (non necessaria per l'Apple II Plus) permette di usare la versione più estesa di BASIC Applesoft.

Auto-Start ROM: (non necessaria per l'Apple II Plus) permette di caricare automaticamente il sistema operativo.

16K Byte Expansion Memory Module: permette di aumentare la memoria sino a 48K.

Hobby/Prototyping Card: permette di costruire da soli le cartoline di interfaccia.

Programmer's Aid 1: (non necessaria per l'Apple II Plus) un insieme di routine in ROM per rendere più semplice e veloce l'Integer BASIC.

Clock/Calendar Card: una cartolina per avere un calendario di 388 giorni ed un orologio con la precisione di 1/1000. Ha una batteria in tampone con l'autonomia di quattro giorni e si possono collegare batterie esterne per aumentare tale periodo.

L'interno del microelaboratore Apple

La fig. 5.3 offre una visione dell'interno con le varie parti che formano l'Apple II. Dovete avere un vostro schermo che potrebbe essere un televisore (bianco e nero o a colori), o un video a parte. La tastiera è simile ad una macchina da scrivere con solo qualche tasto in più. Poiché la parte ester-

na è familiare, la maggior parte di questo paragrafo è dedicata all'interno dell'elaboratore.

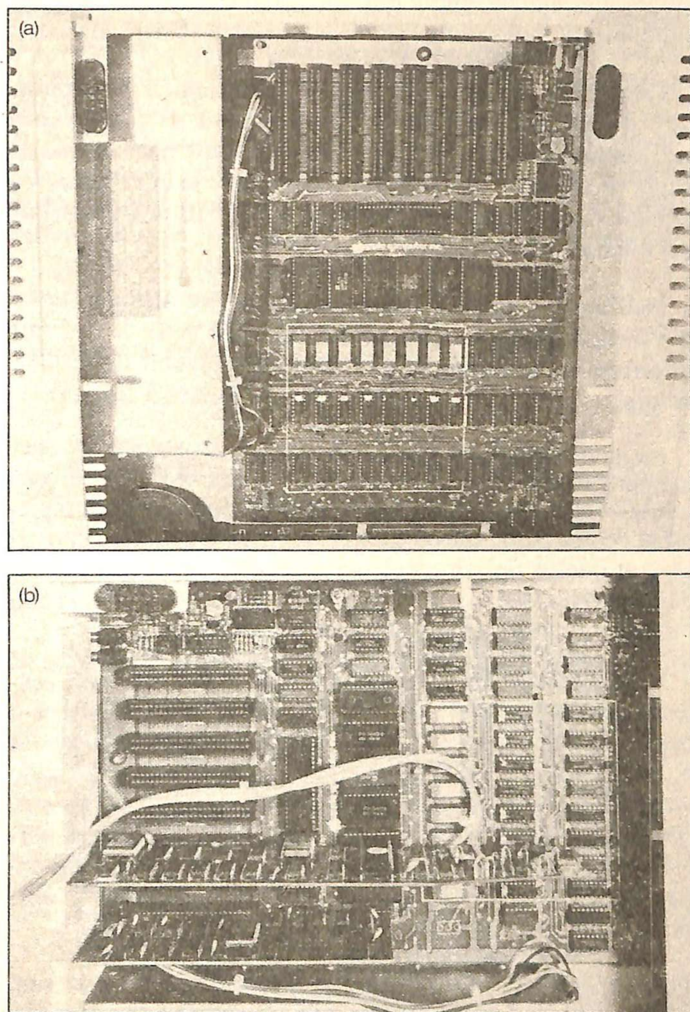


Fig. 5.3 (a) L'interno dell'Apple II (b) l'interno mostra la cartolina di interfaccia

Nell'interno dell'Apple si trovano le parti elettroniche che producono le immagini, i circuiti che producono il suono, la memoria e, naturalmente, tutta la logica per far girare un sofisticato elaboratore. Tutto ciò è montato su un circuito stampato di medie dimensioni, noto come scheda master.

Un circuito stampato è il modo più conveniente per montare e collegare fra loro la gran quantità di componenti dell'elaboratore. Ha su di sé circuiti di rame per collegare i supporti nei quali sono inseriti i vari chip e le interfacce. Lo schema del circuito stampato mostra la struttura essenziale del microelaboratore.

Nell'interno a sinistra (guardando la fine della tastiera) c'è il trasformatore. Esso trasforma i 220 volts di corrente alternata nei voltaggi richiesti dalle componenti dell'elaboratore.

La memoria disponibile per l'utente, particolarmente per memorizzare programmi BASIC, è fornita dai chips di memoria ad eccesso casuale, o RAM. Essi sono posti più o meno al centro della scheda master. L'informazione memorizzata in questo tipo di memoria può esser presa e sostituita dal programma. Quando si spegne l'elaboratore, si perdono tutte le informazioni memorizzate in RAM.

Naturalmente ci sono caratteristiche dell'Apple che vengono sempre richieste e che non possono esser sostituite o perse quando si spegne l'elaboratore. Facciamo due esempi: il BASIC deve esser sempre utilizzabile; i caratteri sullo schermo dovrebbero potersi formare in qualsiasi momento. Questi compiti sono assolti dai chips, con informazioni permanentemente memorizzate in essi. Tali chips sono noti come «memorie di sola lettura» o ROM. Le ROM dell'Apple si trovano proprio sopra le RAM.

Dietro la scheda master si trovano gli zoccoli per collegare l'Apple agli altri dispositivi. Ci sono sette zoccoli per interfacce. Tra questi e le ROM c'è il processore (il chip più grande sulla scheda).

L'uso dell'Apple come cronometro

Leggendo altri libri di programmazione in BASIC, potreste incontrare un comando chiamato TI o TI$. Tale comando ordina all'elaboratore di mostrare il valore di un contatore interno che parte automaticamente quando si accende l'elaboratore. Il calcolo del tempo di questo contatore è molto preciso e perciò ci si riferisce a lui come ad un orologio. Gli elaboratori che hanno questo contatore/orologio possono essere usati come un cronometro complesso e preciso. Purtroppo il contatore/orologio dell'Apple è un accessorio a richiesta e potreste non averlo nel vostro elaboratore. Per provare digitate il comando:

TI<RETURN>

Potreste vedere visualizzata questa risposta:

```
? SYNTAX ERROR
]
```

Questo paragrafo mostrerà che con un po' di immaginazione l'Apple può simulare un cronometro di buona precisione. La routine potrà essere inserita facilmente in un pro-

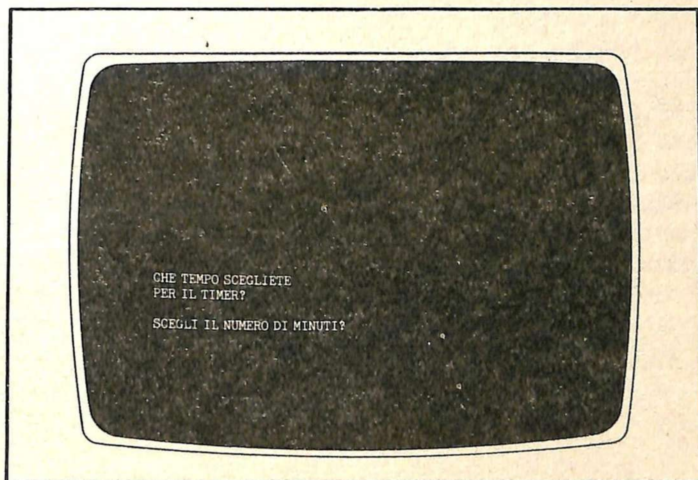


Fig. 5.4 Cosa appare sul video con il programma «Timer»?

gramma di ricette. Come sapete le ricette dicono sia gli ingredienti, le istruzioni, sia il tempo di cottura. Potete chiamare il seguente programma come una subroutine (usando GOSUB e RETURN) adatta ad essere un cronometro per ricette. Digitate il programma e provate a vedere cosa fa. Non vi sono nuovi comandi, ma osservate che usiamo un punto interrogativo al posto del comando PRINT. «?» è più rapido da digitare e prende meno spazio nella riga: è semplicemente un'abbreviazione BASIC di PRINT.

```
 10 HOME
 20 ? "CHE TEMPO SCEGLIETE"
 30 ? "PER IL TIMER?"
 40 ?
 50 ? "SCEGLI IL NUMERO DI MINUTI"
 60 INPUT A
 70 HOME: ? "CONTA FINO A(SPC)";A;"(SPC)
MINUTI"
 80 FOR I = 1 TO A: FOR J = 1 TO 42600: NEXT J
 90 HOME:? "CONTA FINO A(SPC)";A;"(SPC)
MINUTI"
100 ?
110 ?I;"(SPC)MINUTI CONTATI SU (SPC)";A
120 X = —16336
130 NEXT I
140 FOR L = 1 TO 300
150 S = PEEK(X): NEXT L
160 HOME
170 ? "CONTATO"
180 ?
190 ? "VUOI AZZERARE IL TIMER? S/N"
200 INPUT B$
210 IF B$ = "S" THEN GOTO 10 ELSE END
```

Questo programma illustra le più semplici istruzioni del BASIC che avete già incontrato: PRINT, FOR...NEXT e GOTO. Ma illustra anche un interessante uso del loop FOR...NEXT, particolarmente utile nei programmi Apple; cioè un «loop di ritardo». Notate il formato del loop:

```
FOR I = 1 TO N: NEXT I
```

Non c'è alcun comando separato tra le parti FOR e NEXT dell'istruzione. Cioè questa costruzione comanda all'elaboratore di non far altro che girare in cerchio per un determinato numero di volte. Variando il numero dei giri potete determinare la durata del tempo necessario all'elaboratore per eseguire il comando e, da qui, la durata del ritardo del programma.

Notate che il nostro loop di ritardo (che è il cronometro) alla riga 80 è da 1 a 42600 incremento + 1. L'esperienza ha mostrato che all'Apple serve quasi esattamente 1 minuto per completare questo numero di giri. Se infatti trovate che sono solo 59,9 secondi, provate ad aggiungere pochi giri: viceversa, se vi sembra che l'Apple ci metta troppo, riducete il numero, diciamo, a 42000.

La riga 80 mostra anche una caratteristica nota come «loop nidificati». Il primo loop è FOR I = 1 TO A. Ora, A, come potete vedere alla riga 60, è una variabile inserita dall'utilizzatore per specificare la durata del ritardo in minuti. Il loop FOR J è ripetuto tante volte quante specifica il loop FOR I.

Le righe 90-130 forniscono un semplice contatore e un segnale acustico per indicare il passaggio del tempo. Provate, cambiando la riga 110, a leggere:

```
110 ? A-I;"(SPC)MINUTI DI DURATA"
```

Conclusione

L'intenzione di questo libro è stata quella di fornire una facile introduzione all'uso dell'elaboratore Apple II; abbiamo descritto molte applicazioni in cui l'Apple può esser usato per caricare e far girare un programma. C'è un vasto numero di applicazioni di questo tipo, comprese molte applicazioni d'affari, che non richiedono alcuna conoscenza di come l'Apple lavori e necessitano solo di una minima esperienza delle istruzioni necessarie. In queste condizioni, l'unica cosa importante è il programma. L'Apple II è unicamente un mezzo per far girare il programma. Uno speciale sistema di applicazione generale di questo tipo può dimostrare il proprio valore, ripagandosi da sé in brevissimo tempo. Co-

munque l'Apple II è estremamente versatile, essendo in grado di svolgere tutte le attività per le quali può esser programmato. Tale versatilità può esser usata facendo girare differenti programmi per ciascuna sfera di applicazione.

Non sempre i programmi acquistati fanno esattamente ciò che volete; è perciò utile essere in grado di programmare l'Apple II per poterli modificare. Sia per queste ragioni, sia come risultato della curiosità su come sfruttare tutta la potenzialità dell'Apple II, è utile saper scrivere i programmi. Questo libro offre un'introduzione alla programmazione dell'Apple II.

In queste pagine, più di una volta, abbiamo evidenziato l'importanza dell'Apple II come strumento educativo. Non se ne può trascurare l'importanza come esempio di moderna tecnologia. È un merito già il solo fatto di essere un utile prodotto della tecnologia che verrà usato sempre di più in futuro, dimostrando come la tecnologia venga applicata alla vita quotidiana.

Visto da diverse prospettive, l'Apple II appare come uno strumento utile in molte occasioni. Con questo libro ci siamo sforzati di introdurre molte di queste utilizzazioni e di indicare le fonti di informazione che vi aiutino a sviluppare queste ulteriori possibilità.

Appendice 1

Particolari dell'Apple IIe

Riportiamo l'annuncio fatto dalla Apple per presentare l'Apple IIe.

ENHANCED APPLE IIe PERSONAL COMPUTER FEATURES
MORE MEMORY,NEW KEYBOARD, LOWER PRICE

London, January 19, 1983 — Apple Computer (UK) Ltd, British subsidiary of Apple Computer, Inc, the company that has sold over 750,000 personal computers since 1977, today announced a new version of its popular Apple II.

Londra, 19 gennaio 1983 — L'Apple Computer (UK), filiale inglese della Apple Computer Inc, l'industria che ha

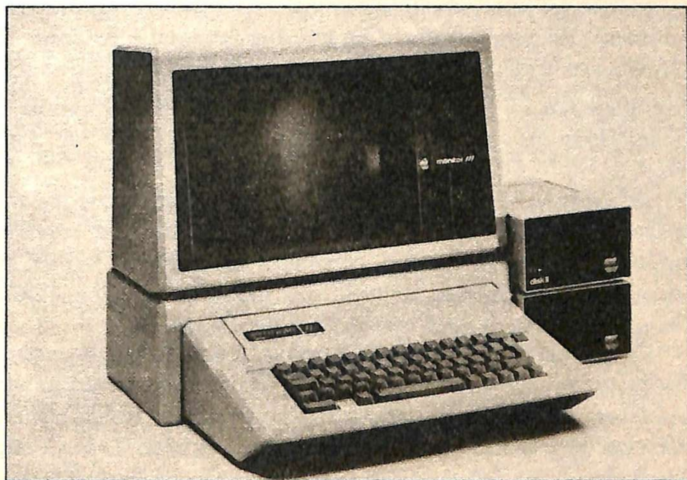


Fig. A1.1 Il nuovo Apple IIe

venduto dal 1977 più di 750.000 computer, ha oggi annunciato una nuova versione del popolare Apple II.

Il nuovo Personal, chiamato Apple IIe, è presentato in tutto il mondo con tastiera in lingua locale e manuali. Il prezzo è vicino a quello dell'Apple II Plus. L'Apple IIe standard ha nuove cartoline logiche, tastiera, contenitori di nuovo disegno ed altri accessori di cui alcuni completamente nuovi per la serie Apple II ed altri disponibili prima solo come accessori a pagamento.

L'Apple IIe sostituirà gradualmente l'Apple II Plus. Tuttavia l'Apple Computer continuerà a fornire accessori e parti di ricambio per i sistemi più vecchi.

«L'Apple IIe è la logica evoluzione del primo Personal», ha detto Keith Hall, direttore del Marketing e delle vendite dell'Apple inglese. «Abbiamo risposto ai desideri della clientela di aggiungere gli accessori più richiesti, lasciando inalterata la possibilità di usare la libreria software dell'Apple II. Con questo annuncio, l'Apple IIe offre possibilità straordinarie e valide per un'ampia gamma di utilizzatori quali professionisti, dirigenti, piccole industrie, educatori, utilizzatori domestici e hobbysti».

Hardware

Ci sono state tredici revisioni dell'Apple II da quando è stato presentato nel 1977 ma nessuna ha l'ampiezza dell'Apple IIe.

Ad esempio l'Apple IIe ha nuove cartoline logiche che usano solo un quarto dei circuiti integrati del più recente modello Apple II. Questa efficienza di progetto significa maggior sicurezza per l'utente: con meno circuiti il sistema consuma meno, scalda meno ed è più affidabile.

Come negli ultimi Apple II, sono inserite 8 porte di espansione di cui una fa risparmiare se si desidera usare un video ad 80 colonne. In un connettore si possono inserire la cartolina per 80 colonne Text Card oppure Extended Text Card, entrambe di nuovo progetto e di costo minore perché molti circuiti sono già inseriti nell'Apple IIe standard.

L'Apple IIe ha di base una memoria RAM di 64 Kilobyte (64K). Si può estendere facilmente a 128K inserendo una cartolina nella Extended Text Card. Più memoria significa poter utilizzare facilmente programmi molto sofisticati come un potente sistema di trattamento di testi, programmi di trattamento dei dati e complicati «data base».

All'esterno l'Apple IIe ha un'ampia tastiera per digitare facilmente, tasti cursore per le 4 direzioni, e maggiore velocità. È anche più facile connettere accessori, quali stampanti e controlli a mano; è anche, per sicurezza, disponibile un coperchio.

Software

In cinque anni sono stati scritti migliaia di programmi software per la serie Apple II. Cioè un'enorme base di programmi software pronti ed immediatamente utilizzabili con il nuovo Apple IIe.

I vecchi programmi non hanno alcun vantaggio dall'Apple IIe, ma: produttori di software — come la Microsoft Corporation con un programma chiamato Multiplan — hanno già elaborato nuove versioni di programmi esistenti e sviluppato nuove applicazioni specifiche per l'Apple IIe.

Mentre l'Apple II si diffonde, l'Apple Computer introduce nuovi programmi in tedesco, francese, inglese e italiano; sono preparati per usare l'Apple IIe, usando anche la memoria aggiuntiva. Si ha una nuova versione, l'Apple Writer II, del programma di trattamento di testi, ed il Quick File II, un nuovo programma versatile che rende facile la ricerca di indirizzi, il calendario ed altri tipi di archivi di lavoro o personali.

Appendice 2

Software per Apple II ed Apple IIe

L'elenco che segue è solo una parte del software disponibile per l'Apple; non è completo e non è una selezione, per cui ottimi pacchetti di software non sono inseriti.

Ammortapple

Gestisce l'ammortamento di tutti i macchinari di un'Azienda.
Informatica Shop

Multiplan per Apple IIe

È un «foglio di lavoro elettronico» di facile impiego che vi consente di risolvere qualsiasi problema che si possa riportare su righe e colonne.
Microsoft

Visi Calc

Sistema di calcolo e gestione dati su matrice di 254 righe e 60 colonne.
VisiCorp™

Visi Plot

Programma per produrre grafici, partendo da una matrice di dati. Produce grafici a linee, a barre, ad area ed a punti.
VisiCorp™

Apple Writer

Programma di elaborazione testi
Apple Computer International

Word Star

È il più potente e raffinato programma di elaborazione testi oggi esistente.
Micropro

Budget finanziario

Calcola lo scostamento partendo dai valori preventivi di bilancio e da quelli consuntivi.
Amministra 2000

Contabilità generale

Programma di contabilità multiaziendale.
Computer Center di Genova

Contabilità semplificata

La procedura permette di ottenere tutte le informazioni necessarie e la valutazione contabile.
Computer System del dott. M. Zisios.

Dichiarazione dei redditi

La procedura consente la stampa e l'archiviazione di tutti i dati del modello 740.
Memor Informatica s.r.l.

Gestione conti correnti bancari

È un insieme di procedure di controllo, di scadenzario e di previsioni del c/c bancario.
Antonelli Alessandro

Fatturazione

Sistema completo di compilazione e stesura delle fatture con emissione di tratta e ricevuta bancaria.
Informatica Biella s.a.s.

Gestione magazzino

Permette la gestione e la valorizzazione fiscale.
Informatica Biella s.a.s.

Gestioni ordini clienti

Programma di contabilità, magazzino e fatturazione.
Adco Informatica

Paghe e stipendi

Programma per gestire sino a 15 tipi di contratti.
Adco Informatica

Visi Calc™ paghe

Calcola e stampa la busta paga dei dipendenti.
Memor Informatica s.r.l.

Apple Logo

Programma che sviluppa il LOGO.
Apple Computer

Pilot

Linguaggio di programmazione per scopi educativi
Apple Computer International™

Magic Spells

Programma per imparare l'ortografia inglese.
Special Delivery Software™

Type attack

Gioco didattico per imparare la dattilografia.
Sirius Software

Assicurazioni

Procedura per Agenzie di Assicurazioni.
Applicazioni Ricerche Informatica s.r.l.

Apple Cash

Si usa l'Apple come registratore di cassa per gestire fino a 4500 articoli.
Computer Center di Genova

Gestione listini

Un programma per gestire listini con codici compresi tra 00.001 e 99.999.
Bul System

Vendita al banco

Il programma permette di ordinare da 1200 a 3200 articoli, a seconda del sistema operativo.
Digital Sync

Dent Apple

Programma per la gestione completa dello studio dentistico.
Informatica Shop

Gestione per medici base

Programmi per la gestione dello studio medico: gestione schedario, compilazione cartella clinica, stampa statistiche e certificati.
Equitec

Vectoranalyzer

Procedura per la rivelazione ed analisi di segnali vettorcardiografici.
La Barbera H & S

A.S.M. Automazione Studio Medico

Gestione completa studio medico.
Emmeprog

Amministrazione condomini

Procedura per la gestione economica del condominio.
Amministra 2000

Programma gestione affitti

Gestione affitti per più appartamenti di più proprietari.
Computeria

Contabilità semplificata per consulenti

Registro IVA, registro vendite, stampa situazione contabile, stampa dichiarazione periodica IVA.
Cigaina

Data Students

Procedura per la segreteria didattica di qualsiasi istituto.
Informatica System

Cofin - Contabilità finanziaria per le scuole

Il programma può gestire tutti gli impegni economici di una scuola.
Informatica System

Apple Hotel

Gestione di tutte le procedure svolte in presenza del cliente.
Data Hotel di Roberto Furco

Gestione Camping

Procedura di gestione delle movimentazioni di un camping con emissione di ricevuta fiscale o fattura.
Sisteda

Score istantaneo per partita di Basket

Si può seguire immediatamente lo score di tutti i giocatori durante la partita.
Centro Computer S.n.c.

Contabilità Lavori Edili

Gestione per: libretto misure, registro contabilità, stati avanzamento lavori, sommario lavori ed emissione certificato di pagamento.
Studio Sistemi Elettronici

Agenzie Immobiliari

Gestione unità immobiliari, ricerca per voci anche registrazione di circa 3500 numeri telefonici.
Amministra 2000

Apple Post

Programma per gestire archivi anagrafici e stampare indirizzi su buste ad etichetta.
Apple Computer International™

Mailing List

Gestisce un archivio di nominativi per l'invio di circolari, bollettini...
Pertel s.n.c Torino

Personal Data Base

Procedura per gestire archiviazione personalizzata senza scrivere una riga di programma.
Iret Informatica

Rubrica Telefonica

Procedure per gestire un archivio di numeri telefonici.
Informatica Shop

Apple Plot

Programma per creare, manipolare, visualizzare e stampare grafici.
Apple Computer International™

Disegno da tastiera

Permette di fare disegni pilotando, da tastiera, un punto elementare.
Iteco

Prospettiva 3D

Procedura per la rappresentazione prospettiva su video e su stampante.
Studio Tecnico Ing. Sauro Agostini

Matematica superiore

Programma per operazioni su funzioni, derivate, integrali, sistema di equazioni.
Studio Softidea

Pronostici Totocalcio

Procedura per sviluppare sistemi integrali tra le 8 e le 216 colonne.
Franco Lentini

Archivio di dischi di musica classica

Programma per gestire un archivio di dischi per autore, genere, tipo di esecutore, titolo di brani.
Strechelli Benedetto

Introduzione simultanea nel mondo

La versione internazionale dell'Apple IIe è stata progettata con speciali alimentatori, carte logiche e tastiere adatte

per la Germania, la Francia, l'Inghilterra e l'Italia. L'uscita per il video ha entrambi gli standard PAL oppure NTSC.

Le tastiere sono conformi allo standard 150 ed hanno sia la lingua locale sia i caratteri americani sullo stesso tasto. Un interruttore facilmente accessibile permette ogni volta di scegliere l'insieme di caratteri desiderati. Sono disponibili 14 tastiere europee comprese quelle svedesi, spagnole, portoghesi ed altre.

Appendice 3

Glossario

Argomento

Comunemente usato per indicare il valore associato ad un comando.

ASCII (caratteri)

ASCII (pronunzia «aschi»). Un codice usato nella maggior parte degli elaboratori per rappresentare 128 caratteri alfanumerici e di controllo. Usa 7 bit per ogni carattere. per esempio il codice ASCII per il carattere A è 1000001.

Assembler (linguaggio)

Un linguaggio simbolico a basso livello che usa memoria invece di istruzioni numeriche proprie del linguaggio-macchina.

Base-Dati

Organizzazione dei dati per ritrovarli più facilmente e più velocemente.

BASIC

Il linguaggio ad alto livello più semplice e più facile da imparare. È un acronimo per Begginner's All-purpose Symbolic Instrution Code.

Binario

Un sistema di numerazione con 2 cifre, «0» e «1»; in ogni numero binario la cifra rappresenta una potenza di 2.

Bit

(Binary digit); cifra binaria; può essere 0 o 1.

Bug

Un errore che impedisce al programma di girare, o di girare correttamente.

Byte

Unità di memorizzazione di 8 bit. Lo stesso di memoria necessaria per un solo carattere. 32K Byte sono circa 32000 locazioni di memoria per altrettanti caratteri.

Cartolina

È detta anche «scheda a circuito stampato». È di materiale plastico ed i circuiti necessari sono stampati sulla superficie. Vi sono anche dei piccoli fori in cui i componenti elettronici possono esser attaccati o saldati, per collegarli al circuito. All'estremità del circuito vi sono i connettori che permettono di collegare tra di loro i vari circuiti, attraverso la scheda master. Le funzioni di elaborazione e la memoria di un microelaboratore sono su alcune cartoline.

Chip

Letteralmente un «pezzetto» di silice, usato per indicare i circuiti integrati.

Circuito integrato

Un circuito elettronico miniaturizzato in un chip di silice di alcuni mm^2.

Codice-Macchina

Il codice compreso dall'elaboratore.

CPU

(Central Processing Unit). Il «cervello» dell'elaboratore che contiene i circuiti elettronici che interpretano ed eseguono le istruzioni.

Cursore

Barra o quadratino lampeggiante che indica la posizione in cui sarà mostrato il carattere successivo.

Debug

Trovare gli errori in un programma.

Default

Il valore assegnato automaticamente da programma.

Densità

Il numero di bit memorizzati per cm^2.

Diagramma di flusso

Un disegno che concisamente individua i passi di un programma. Si usa come aiuto nel programmare.

Disco

Supporto per memorizzare le informazioni.

Doppia densità

Densità doppia rispetto alla singola.

Doppia faccia

Quando i dati sono memorizzati su entrambi i lati del dischetto.

DOS (Disk Operating System)

Sistema operativo per i drive. Un programma che facilita la scrittura e la lettura del disco. Il DOS sceglie la porzione di disco in cui memorizzare i dati e ricorda dove siano per leggerli.

Drive

Unità periferica per registrare e leggere dati sul dischetto.

Eseguire

Obbedire ad una istruzione di un programma. Sinonimo di «far girare» il programma.

Faccia

Lato del dischetto in cui sono memorizzate le informazioni.

Grafici

Disegni fatti dall'elaboratore.

Hardware

È l'insieme di parti fisiche che formano un elaboratore.

Indirizzo

La locazione dove è memorizzata l'informazione, nella memoria dell'elaboratore, nel nastro della cassetta, nel dischetto.

Inizializzazione

Caricare il sistema operativo da un disco o da un nastro nella memoria dell'elaboratore.

Interfaccia

Connessione elettronica e/o fisica tra due unità. L'interfaccia seriale trasmette o riceve un bit alla volta, l'interfaccia parallelo trasmette o riceve più bit alla volta.

Interprete

Il programma che traduce un linguaggio ad alto livello, nel linguaggio-macchina.

K

Abbreviazione per kilobyte di memoria; un K = 1024 byte.

Linguaggio ad alto livello

Linguaggio di programmazione più comprensibile del linguaggio-macchina. Ad esempio BASIC, Pascal, FORTRAN.

Linguaggio a basso livello

Linguaggio più comprensibile alla macchina che all'uomo; per esempio, assembler.

Listato

La stampa o la presentazione, sul video, delle righe di istruzioni di programma.

Memoria

Memoria principale sono i circuiti in cui la CPU memorizza direttamente le informazioni. Memoria periferica sono i supporti in cui la CPU memorizza attraverso le interfacce.

Microelaboratore

Un elaboratore in cui l'unità centrale è un microprocessore.

Microprocessore

Circuito integrato molto complesso che può esser programmato per svolgere diverse attività.

Modalità

Un insieme di condizioni o di regole da applicare.

Output

Informazioni presentate sulle periferiche.

Periferica

Unità collegate all'elaboratore che le può controllare ed usare; ad es. drive, video e stampante.

Porta

Il connettore dell'elaboratore in cui possono essere inseriti i terminali delle periferiche.

Programma

Successione di comandi dati all'elaboratore, che li esegue automaticamente.

RAM

Random Access Memory: memoria il cui contenuto è perso quando si toglie l'alimentazione.

ROM

Read Only Memory: memoria permanente, usata per memorizzare le informazioni di continuo uso; ad es. il BASIC.

Scroll

Il muoversi avanti e indietro delle informazioni sul video.

Singola densità

Standard di memorizzazione dei dati.

Sistema operativo

Software che controlla l'elaboratore e le periferiche.

Software

Programmi per l'elaboratore.

Stringa

Un insieme di parole o caratteri registrati.

Trattamento dei testi

Un programma per scrivere, correggere, memorizzare e stampare i testi.

Vettore

Un insieme di dati identificati sotto lo stesso nome da un indice. Così il comando DIM A$(20) assegna 20 locazioni di memoria al vettore che può esser esaminato in sequenza con i nomi A$(1), A$(2)...

Appendice 4

Un gioco: La Torre di Hanoi

Questo gioco è un classico e antico problema di logica. Avete una pila di dischi di diverso diametro e potete spostare un disco alla volta dal punto A al punto C col minor numero di mosse. Per aiutarvi a spostare i dischi, c'è il punto B che è una posizione temporanea. La difficoltà, tuttavia, consiste nel non poter mettere un disco sopra ad un altro più piccolo.

Per usare il programma, digitate con attenzione il seguente listato, digitate RUN e seguite le istruzioni che appaiono sullo schermo. Apparirà subito a sinistra una torre della grandezza voluta. I punti B e C, all'inizio vuoti, sono mostrati a destra. Per spostare i dischi, digitate semplicemente il punto in cui è il disco da muovere (ad es. A), seguito dal

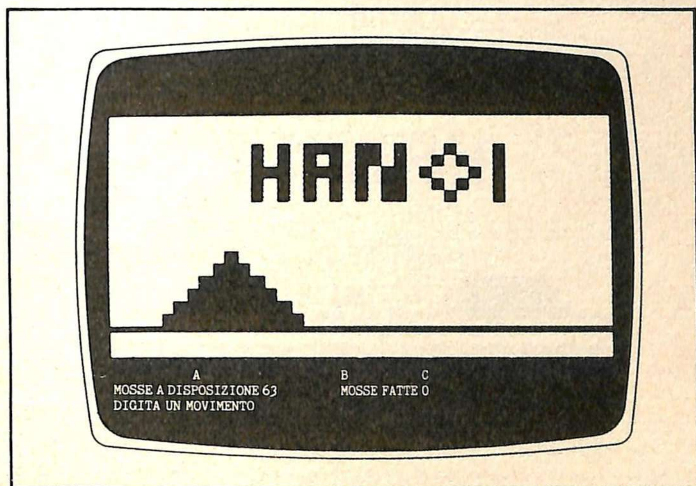


Fig. A4.1 La torre di Hanoi - inizio

punto in cui lo volete spostare (per es. C). Così «A,C» sarà la vostra prima mossa. Buon divertimento.

```
10 REM      LA TORRE DI HANOI
11 REM
19 REM      INTRODUZIONE ED ISTRUZIONI
20 HOME: PRINT: PRINT: PRINT TAB(12);"TORRE
DI HANOI"
30 PRINT: PRINT: PRINT "ALLORA:"TAB(60);"UN
GRUPPO DI DISCHI FORMA UNA TORRE": PRINT
TAB(10)"NELLA POSIZIONE A. DOVETE"
40 PRINT TAB(10)"SPOSTARLA NELLA POSIZIO
NE C"
50 PRINT: PRINT"REGOLA 1:" TAB(10) "SPOSTA
TE UN DISCO ALLA VOLTA": PRINT TAB(10)"OGNI
VOLTA"
60 PRINT: PRINT"REGOLA 2:" TAB(10)"UN DISCO
GRANDE NON PUÒ": PRINT TAB(10)"STARE SU UN
DISCO PICCOLO"
```

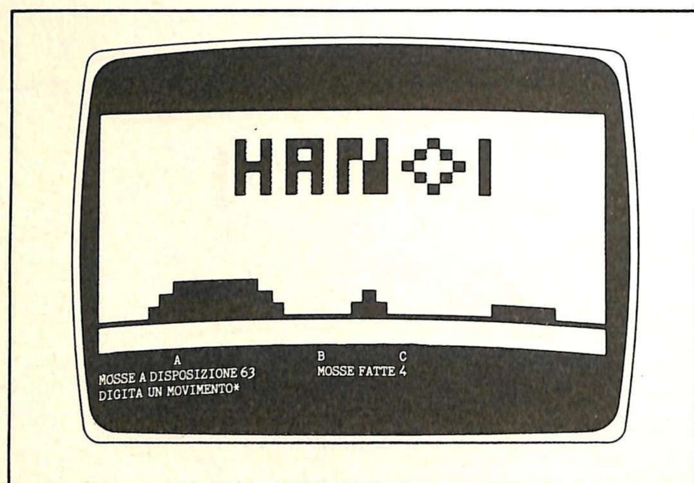


Fig. 4.2 La torre di Hanoi - dopo alcune mosse

```
 70 PRINT: PRINT"INIZIO:" TAB(10)"PER MUOVE
RE SPINGERE I TASTI A,B e C"
 75 FLASH
 80 PRINT: PRINT: PRINT: PRINT"SPINGI UN TA
STO PER INIZIARE"
 90 GET A$: IF A$ = ""THEN 90
100 HOME: NORMAL: M = 0
110 PRINT: PRINT: PRINT TAB(10)"TORRE DI HA
NOI"
120 PRINT: PRINT: PRINT"QUANTI DISCHI VUOI?"
130 PRINT: PRINT"SCEGLI UN NUMERO TRA 2 E 6"
140 GET A$: IF A$ = ""THEN 140
150 C = VAL(A$)
160 IF C>1 AND C<7 THEN 190
170 FLASH: PRINT: PRINT: PRINT: PRINT"HAI DIGI
TATO UN NUMERO NON VALIDO": PRINT: PRINT"RI
PROVA"
180 FOR I = 1 TO 1500: NEXT I: GOTO 100
190 NORMAL: PRINT: GR
200 FOR I = 1 TO 6: READ A(I,1): A(1,2) = 0: A(I,3) = 0:
NEXT I
210 DATA 1,2,3,4,5,6
220 H(1) = C: H(2) = 0: H(3) = 0
230 COLOR = 13
240 FOR I = 0 TO 39: VLIN 0,39 AT I: NEXT I
250 R = 5: COLOR = 8
260 FOR I = 1 TO 5: READ N
270 FOR J = 1 TO N: READ X: PLOT 10 + X,R: PLOT
10 + X,R + 1: NEXT J
280 R = R + 2: NEXT I
290 DATA 10,1,3,5,6,7,9,10,12,16,20
291 DATA 10,1,3,5,7,9,10,12,15,1,7,20
292 DATA 12,1,2,3,5,6,7,9,11,12,14,18,20
293 DATA 10,1,3,5,7,9,11,12,15,17,20
294 DATA 9,1,3,5,7,9,11,12,16,20
400 Z1$ = "(SPC × 7)A(SPC × 11)B(SPC × 11)"
410 Z2$ = "MOSSE A DISPOSIZIONE (SPC)" +
STR$(INT(2∧C—1))
420 Z3$ = "MOSSE FATTE"
```

```
 430  Z4$ = "DIGITA UN MOVIMENTO"
 440  Z5$ = "(SPC x 3)"
 450  Z6$ = "MOSSA NON VALIDA—RIPROVA"
 500  GOSUB 5000
 510  PRINT: PRINT Z1$: PRINT Z2$; TAB(20) Z3$;M:
PRINT Z4$: PRINT Z5$;
 520  GOSUB 6000
 525  PRINT: PRINT Z1$: PRINT Z2$;TAB(20)Z3$;M:
PRINT Z5$: PRINT Z4$;
 530  IF H(I1)<1 THEN PRINT Z1$: PRINT Z2$;
TAB(20)Z3$;M: PRINT Z6$: PRINT Z4$: GOTO 520
 540  IF H(I2) = 0 GOTO 560
 550  IF A(C + 1—H(I1),I1)>A(C + 1—H(I2),(I2)THEN
PRINT: PRINT Z1$: PRINT Z2$; TAB(20)Z3$;M: PRINT
Z6$: PRINT Z4$;: GOTO 520
 560  H(I2) = H(I2) + 1
 570  A(C + 1—H(I2),I2) = A(C + 1—H(I1),I1)
 580  A(C + 1—H(I1),I1) = 0
 590  H(I1) = H(I1)—1
 620  GOSUB 5000
 630  IF H(3)<>C THEN GOTO 520
 640  PRINT: PRINT TAB(15)"CONGRATULAZIONI"
 650  PRINT Z2$: TAB(20)Z3$;M: PRINT"SPINGI UN
TASTO PER UNA NUOVA PARTITA"
 660  GET A$: IF A$ = ""THEN 660
 670  TEXT: RUN 100
5000 REM       PRINT ROUTINE
5010 FOR I7 = 1 TO 3
5020 P = 3 + (I7—1)*12: Y = 35—2*C
5030 FOR J1 = 1 TO C
5040 FOR J2 = 0 TO 1: COLOR = 13
5050 IF A(J1,I7) = 0 THEN HLIN P,P + 11 AT Y + J2:
GOTO 5080
5060 HLIN P,P + 6—A(J1,I7) AT Y + J2
5070 COLOR = 1: HLIN P + 7—A(J1,I7),
P + 5 + A(J1,I7) AT Y + J2
5080 NEXT J2
5090 Y = Y + 2
```

```
5100 NEXT J1,I7
5110 COLOR = 1: HLIN 0,39 AT 35
5120 RETURN
6000 REM       DIGITA UNA MOSSA M = M + 1
6010 GET A$: IF A$ = ""THEN 6010
6030 GET B$: IF B$ = ""THEN 6030
6040 I1 = ASC(A$)—64: I2 = ASC(B$)—64
6060 IF I1>0 AND I1<4 AND I2>0 AND I2<4 THEN
RETURN
6070 PRINT: PRINT Z1$: PRINT Z2$;TAB(20)Z3$;M
6080 PRINT Z6$: PRINT Z4$
6090 GOTO 6010
```

Lire 16000